

AD-A086 067

HARRIS CORP MELBOURNE FL GOVERNMENT COMMUNICATION SY--ETC P/8 9/2
MULTIPLE MICROPROCESSOR SYSTEM (MMS) DESIGN STUDY. VOLUME IV.(U)
MAR 80 F30602-78-C-0114
NL

UNCLASSIFIED

RADC-TR-80-33-VOL-4

NL

Page 1

20

AD-A086 067

1

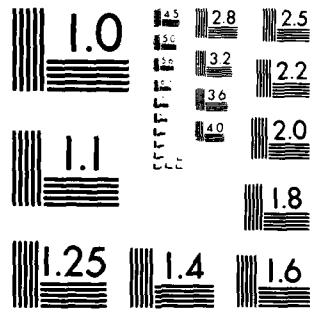
END

DATE

FILED

8-80

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ADA 086067

DDC FILE COPY

RADC-TR-80-33, Vol IV (of four)
Final Technical Report
March 1980

LEVEL
12
A086066
Rome Air Development Center

MULTIPLE MICROPROCESSOR SYSTEM (MMS) DESIGN STUDY

Harris Corporation

Government Communications Systems Division

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DTIC
ELECTE
S JUL 1 1980 D
A

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

80 6 30 023

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-80-33, Volume IV (of four) has been reviewed and is approved for publication.

APPROVED: *Michael A. Troutman*
MICHAEL A. TROUTMAN, 1Lt, USAF
Project Engineer

APPROVED: *Wendall C. Bauman*
WENDALL C. BAUMAN, Colonel, USAF
Chief, Information Sciences Division

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DDC TAB	
Unannounced	
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

FOR THE COMMANDER:

John P. Huss
JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (ISCA) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER	
18 RADC-TR-80-33-VOL 4 (of four)	AD-A086067		
4. TITLE (and Subtitle)		5. TYPE OF REPORT & PERIOD COVERED	
6 MULTIPLE MICROPROCESSOR SYSTEM (MMS) DESIGN STUDY. Volume IV.		9 Final Technical Report.	
7. AUTHOR(s)		6. PERFORMING ORG. REPORT NUMBER	
Harris Corporation Government Communications Systems Division		N/A	
9. PERFORMING ORGANIZATION NAME AND ADDRESS		8. CONTRACT OR GRANT NUMBER(s)	
Harris Corporation Government Communications Systems Division, P O Box 37, Melbourne FL 32901		15 F30602-78-C-0114	
11. CONTROLLING OFFICE NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
Rome Air Development Center (ISCA) Griffiss AFB NY 13441		16 63728F 25290104	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE	
Same		11 March 1980	
		13. NUMBER OF PAGES	
		91	
		15. SECURITY CLASS. (of this report)	
		UNCLASSIFIED	
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
		N/A	
16. DISTRIBUTION STATEMENT (of this Report)			
Approved for public release; distribution unlimited.			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
Same			
18. SUPPLEMENTARY NOTES			
RADC Project Engineer: 1Lt Michael A. Troutman (ISCA)			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)			
multiple microprocessors		microprocessor	
emulation		distributed architecture	
computer architecture		performance measurement	
architecture evaluation		total system design	
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)			
This is a design for a Multiple Microprocessor System (MMS) to be built as part of the System Architecture Evaluation Facility (SAEF) being developed at RADC. The MMS was designed to be used in the modeling of a wide range of multiprocessor configurations for the purpose of evaluating their suitability to unique Air Force data processing requirements.			

DD FORM 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Introduction

This volume of the SAEF FINAL REPORT contains Appendix J and Appendix K. Appendix J, the MMS Hardware Specifications, is contained in pages J1 to J-39. Appendix K, the MMS Software Specifications, is contained in pages K1 to K-40.

APPENDIX J

The MMS Hardware Specifications

TABLE OF CONTENTS

Paragraph	Title	Page
1.0	SCOPE	J1
1.1	<u>Scope</u>	J1
2.0	APPLICABLE DOCUMENTS	J1
2.1	<u>Government Documents</u>	
2.2	<u>Non-Government Documents</u>	J1
3.0	REQUIREMENTS	J2
3.1	<u>System Definition</u>	J2
3.1.1	<u>General Description</u>	J2
3.1.2	Missions	J5
3.1.3	Threat	J6
3.1.4	System Diagrams	J6
3.1.5	Interface Definitions	J6
3.1.5.1	Internal Subsystem Interfaces	J6
3.1.5.1.1	BIU Interface	J6
3.1.5.1.1.1	Functional	J6
3.1.5.1.2	Intersegment BIU	J11
3.1.5.1.2.1	Functional	J11
3.1.5.1.3	Facilities Control Processor Interface	J12
3.1.5.1.3.1	Functional	J12
3.1.5.1.4	Performance Monitoring System Interface	J12
3.1.5.1.4.1	Functional	
3.1.5.1.5	I/O Processor Interface	J12
3.1.5.1.5.1	Functional	J12
3.1.5.2	External Subsystem Interfaces	J13
3.1.5.2.1	ARPANET Interface	J13
3.1.5.2.1.1	Functional	J13
3.1.5.2.2	Local Terminal Interfaces	J13
3.1.5.2.2.1	Functional	J13
3.1.6	Government Furnished Property List	J14
3.1.7	Operational and Organizational Concepts	J14
3.2	<u>Characteristics</u>	J15
3.2.1	<u>Performance Characteristics</u>	J15
3.2.1.1	Emulation Efficiency	J15
3.2.1.2	Communications Bus Subsystem	J18
3.2.1.3	Emulated I/O	J18
3.2.1.4	Shared Resource Controller Subsystem	J19
3.2.1.5	Time Alignment Controller Subsystem	J19
3.2.1.6	PMS	J19
3.2.2	Physical Characteristics	J19
3.2.2.1	Physical Location	J19
3.2.2.2	Clearance Envelope	J19
3.2.2.3	Mobility	J19

Paragraph	Title	Page
3.2.2.4	Size	J19
3.2.2.5	Weight	J19
3.2.2.6	Noise Generation	J20
3.2.2.7	Grounding	J20
3.2.2.8	Security	J20
3.2.3	Reliability	J20
3.2.4	Maintainability	J20
3.2.5	Availability	J20
3.2.6	System Effectiveness Models	J20
3.2.7	Environmental Conditions	J20
3.2.7.1	Temperature	J20
3.2.7.2	Humidity	J20
3.2.7.3	Operating Voltage	J21
3.2.7.4	Dust and Smoke	J21
3.2.8	Nuclear Control Requirements	J21
3.2.9	Transportability	J21
3.3	<u>Design and Construction</u>	J21
3.3.1	Materials, Processes, and Parts	J21
3.3.2	Electromagnetic Radiation	J21
3.3.3	Identification and Marking	J21
3.3.4	Workmanship	J22
3.3.5	Interchangeability	J22
3.3.6	Safety	J22
3.3.7	Human Performance/Human Engineering	J22
3.4	<u>Documentation</u>	J22
3.5	<u>Logistics</u>	J23
3.5.1	Maintenance	J23
3.5.2	Supply	J23
3.5.3	Facilities and Facility Equipment	J23
3.6	<u>Personnel and Training</u>	J23
3.6.1	Personnel	J23
3.6.2	Training	J23
3.7	<u>Functional Area Characteristics</u>	J23
3.7.1	Communications Bus System	J23
3.7.1.1	SBS	J23
3.7.1.2	BSC	J24
3.7.1.3	BIU	J25
3.7.1.4	Intersegment Bus	J25
3.7.2	Processing Element Subsystem	J25
3.7.2.1	Instruction Execution Unit	J26
3.7.2.2	Local Memory	J28
3.7.2.3	Memory Interface	J29
3.7.2.4	Message Arbitration Logic	J31
3.7.3	Broadcast Communication Controller Subsystem	J31
3.7.3.1	Segment Broadcast Controller	J32
3.7.3.2	Master Broadcast Controller	J32

Paragraph	Title	Page
3.7.4	Time Alignment Controller Subsystem	J33
3.7.4.1	TAC	J33
3.7.4.2	LPA	J33
3.7.5	Emulated Shared Resource Controller Subsystem	J34
3.7.6	Facilities Control Processor Subsystem	J34
3.7.6.1	FCP	J34
3.7.6.2	FCP Interface	J34
3.7.6.3	PMS Interface	J35
3.7.6.4	IOP Interface	J36
3.7.6.4.1	IOP Ports	J36
3.7.6.4.2	IPC Buffer	J36
3.7.6.4.3	Interval Timers	J36
3.7.6.5	ARPANET Interface	J36
3.7.6.6	Local Terminal Interfaces	J37
4.0	QUALITY ASSURANCE PROVISIONS	J37
4.1	<u>General</u>	J37
4.1.1	Responsibility for Inspection	J37
4.2	<u>Quality Conformance Inspection</u>	J38
5.0	PREPARATION FOR DELIVERY	J39
6.0	APPENDIX	J33
6.1	Appendix A, Glossary of Acronyms and Abbreviations	J38

1.0 SCOPE

1.1 Scope

This specification establishes the performance, design, development, test requirements, and qualifications of Multiple Micro-processor System Emulation equipment, hereinafter referred to as the MMS.

2.0 APPLICABLE DOCUMENTS

2.1 Government Documents

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referred herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

Specifications:

Document Number	Title	P.N.
NTIS AD-055996	Specification for the Inter-connection of a Host and and IMP.	3.1.5.2.1.1
NTIS AD-A052594	ARPANET Protocol Handbook	3.1.5.2 1.1

Standards:

Document Number	Title	P.N.
Military Standard 454 Requirement 9	Standard General Requirements for Electronic Equipment	3.3.4

2.2 Non-Government Documents

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

Specifications:

Document Number	Title	P.N.
Contract No. F30602-78-0114	SAEF Final Report Vol. 1, November 1979	3.2.1
6277-7751-004	Preliminary Hazard Report Analysis for SAEF/MMS	3.3.6

Standards:

Not applicable.

3.0 REQUIREMENTS

3.1 System Definition

The MMS is a multiprocessor system which provides Rome Air Development Center (RADC) with the capability to emulate a wide variety of multiprocessor systems under user control. These emulated systems shall include distributed systems employing master/slave and distributed control philosophies. The MMS shall effectively emulate Single Instruction Multiple Data (SIMD), Multiple Instruction Single Data (MISD), and Multiple Instruction Multiple Data (MIMD), as well as Single Instruction Single Data (SISD) architectures. The MMS shall also have the capability to be interfaced to the already existing computer system at the System Architectural Evaluation Facility (SAEF) at RADC.

3.1.1 General Description

The MMS shall provide a single user with the capability to accurately emulate a wide variety of multiple processor systems the MMS shall contain a minimum of 64 Processing Elements (PE's). These PE's shall consist of a 16-bit microprogrammable computer and all the required support and interconnect logic. To achieve an accurate emulation, the MMS shall

contain additional hardware to perform the functions of emulated I/O control, emulated shared resource control, time alignment control of the emulation, broadcast communications control, and collection of data to measure the performance of the system being emulated. To interconnect the MMS PE's and control hardware, MMS shall contain a highly flexible and versatile interelement interconnect structure which facilitates communication between all MMS elements. The MMS shall be divided into the following functional subsystems:

- a. a Communications Bus Subsystem
- b. a Processing Element Subsystem
- c. a Time Alignment Controller Subsystem
- d. a Broadcast Communications Controller Subsystem
- e. an Emulated Shared Resource Controller Subsystem
- f. a Facilities Control Processor Subsystem

The subsystem functions and their major components are as follows:

a. The Communications Bus Subsystem - provides a high bandwidth communication channel for transfer of various types of information between the various MMS subsystems listed above. The Communications Bus Subsystem consists of:

- (1) a segmented high bandwidth synchronous bus structure (SBS),
- (2) bus segment controller (BSC) units for each bus segment (1 each),
- (3) bus interface units for each device connected to the various SBS segments (1 each), and
- (4) intersegment bus interface units for each bus segment which houses PE's (1 each)

b. The Processing Element Subsystem - provides a microprogrammable PE which can be user programmed to emulate a wide variety of micro and minicomputers. The Processing Element Subsystem consists of:

- (1) an instruction execution unit (IEU).
- (2) a local memory unit (LM).
- (3) a memory interface unit (MI).
- (4) message arbitration logic between the IEU, LM, MI, and the BIU.

c. The Time Alignment Controller Subsystem - provides the MMS user with the capability to time align the 64 PE's in MMS. The Time Alignment Controller Subsystem consists of:

- (1) a Master Time Alignment Controller Unit (TAC).
- (2) Local Pseudo-Time Accumulator Units (LPA) for each PE (1 each).

d. The Broadcast Communications Controller Subsystem - provides the capability to emulate single instruction multiple data (SIMD) and/or multiple instruction single data (MISD) machines. The Broadcast Communications Controller Subsystem consists of:

- (1) a Master Broadcast Controller Unit,
- (2) Segment Broadcast Controller Units for each SBS segment which houses PE's (1 each).

e. The Emulated Shared Resource Controller Subsystem - provides the MMS with the capability of determining when a PE can or cannot access an emulated shared resource. The Emulated Shared Resource Controller consists of:

- (1) a PE specially programmed to control all PE request to access emulated shared resources.

f. The Facilities Control Processor Subsystem - provides the functions of system initialization, collection of performance monitoring system (PMS) data, control of emulated I/O (IOP) and interfacing MMS to external systems. The Facilities Control Processor Subsystem consists of:

- (1) a Facilities Control Processor (FCP).
- (2) an FCP interface unit to SBS.
- (3) a PMS interface unit to SBS.
- (4) an IOP interface unit to SBS.
- (5) an ARPANET interface unit to FCP.
- (6) local terminal interface unit to FCP.

3.1.2 Missions

The mission of the MMS is to accurately and efficiently model a wide variety of real time multiple processor subsystems. To perform a true emulation, the MMS must be capable of directly executing the software and/or firmware (without modification) of any system being emulated. To achieve an accurate emulation the MMS must allow the user insight into a process without perturbing the process. To achieve this goal the MMS must provide a mechanism which maintains the time coherence associated with the real process. To achieve an efficient emulation the performance monitoring of an emulated system should not effect (reduce) the emulation efficiency of the MMS when performance data is not being collected. The MMS must also provide a means of modeling the timeouts, time delays, error detection, bus arbitration, and bus protocol which may occur in the system to be emulated. The achievement of this mission requires the MMS to contain a highly flexible and versatile Processing Element Subsystem. In addition, the MMS must

also provide the capability to simulate environmental data, emulate I/O data and emulate shared resources.

3.1.3 Threat

Not applicable.

3.1.4 System Diagrams

The basic system diagrams are illustrated in Figures 3.1.3.1 to 3.1.4.4. These figures illustrate the overall system and how the system is physically segmented. SBS segments one through four are used to house 64 PE's in MMS. The fifth segment is used as the central communications segment for the inter-segment communications between the MMS control element and the PE segments. Figure 3.1.4.4 provides a more detailed diagram of the functional areas of the Processing Element Subsystem.

3.1.5 Interface Definitions

3.1.5.1 Internal Subsystem Interfaces

The interconnection of subsystems within the MMS system requires the following interfaces:

- a. Bus Interface Units (BIU's).
- b. Intersegment Bus Interface Units.
- c. FCP interface.
- d. PMS interface.
- e. IOP interface.

3.1.5.1.1 The BIU Interface

3.1.5.1.1.1 Functional

Every device in MMS requires a BIU in order to communicate over the SBS busing structure. The BIU shall perform all the necessary handshaking required by SBS protocol. The characteristics for each BIU are as follows:

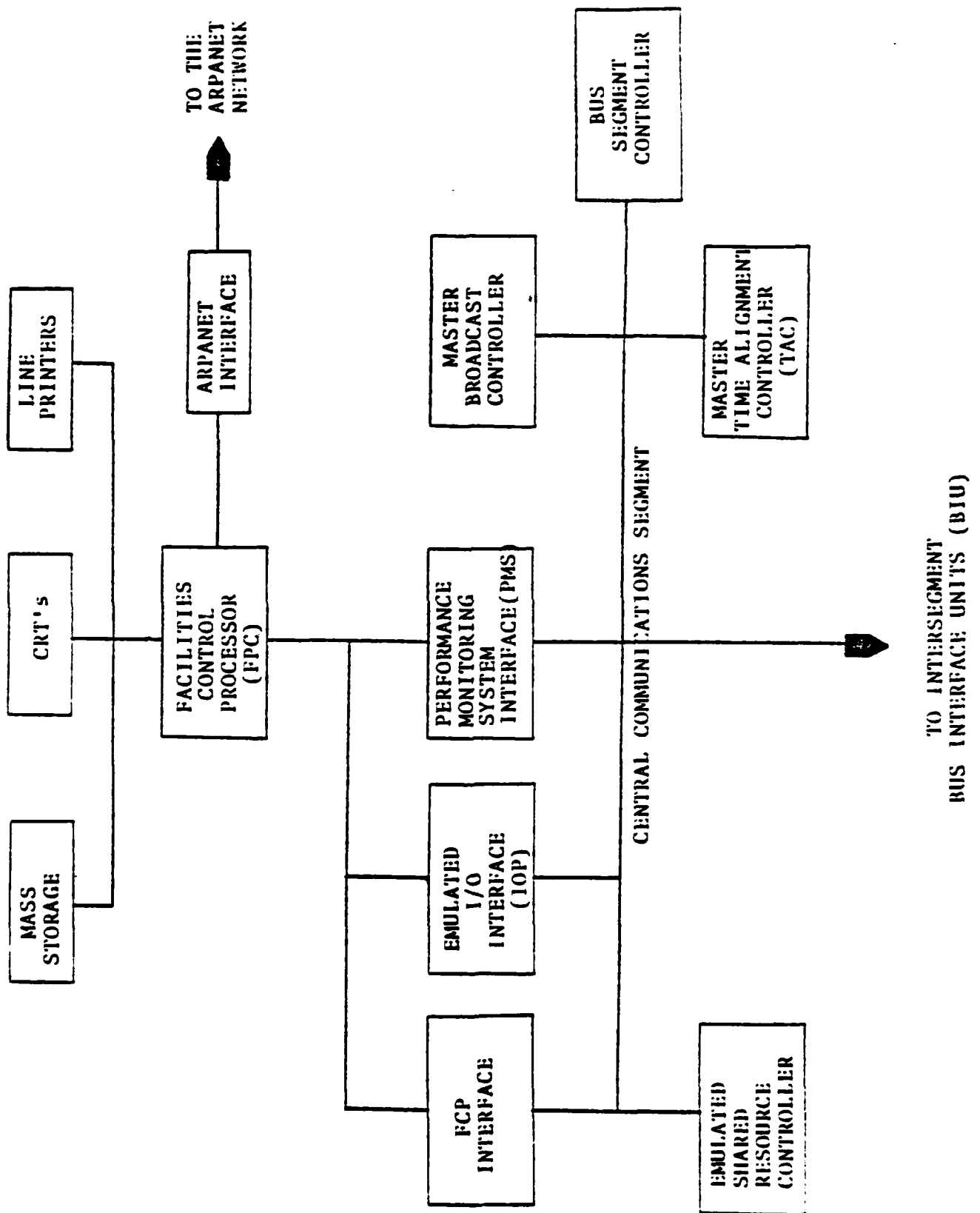


Figure 3.1.4.1 THE CENTRAL COMMUNICATIONS SEGMENT

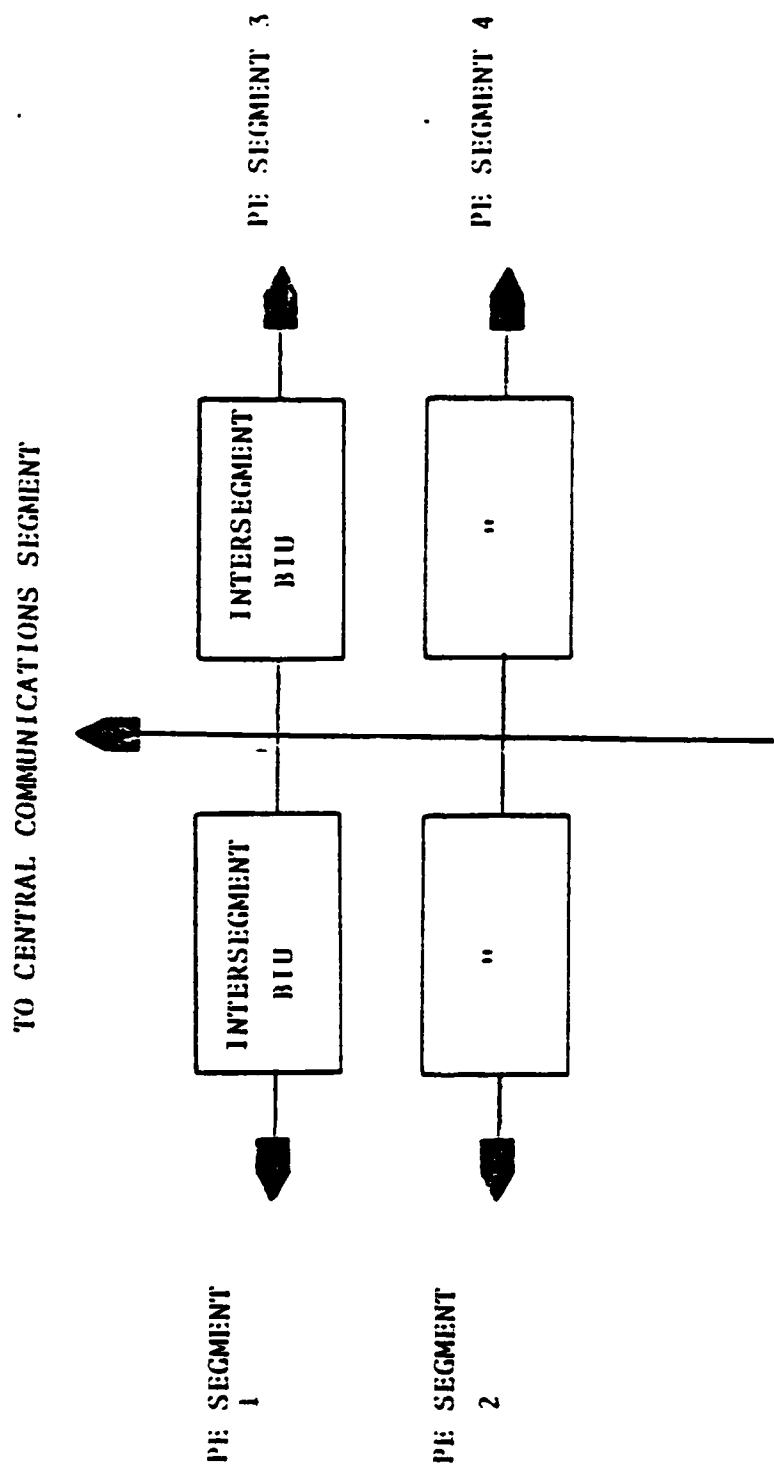


Figure 3.1.4.2 INTERCONNECTION OF SEGMENTS

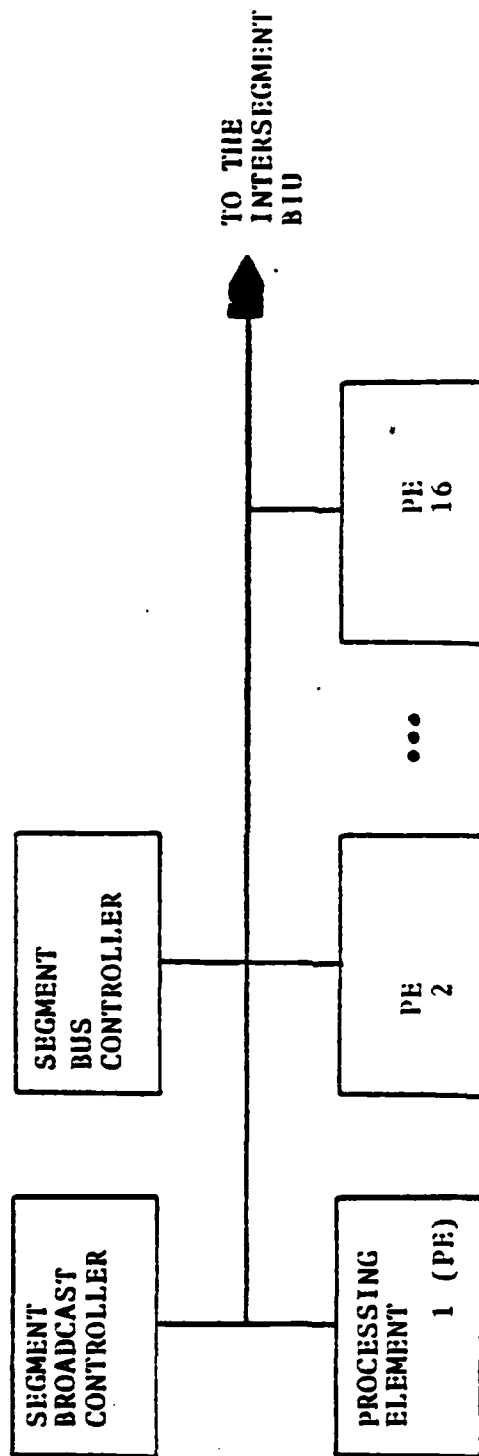
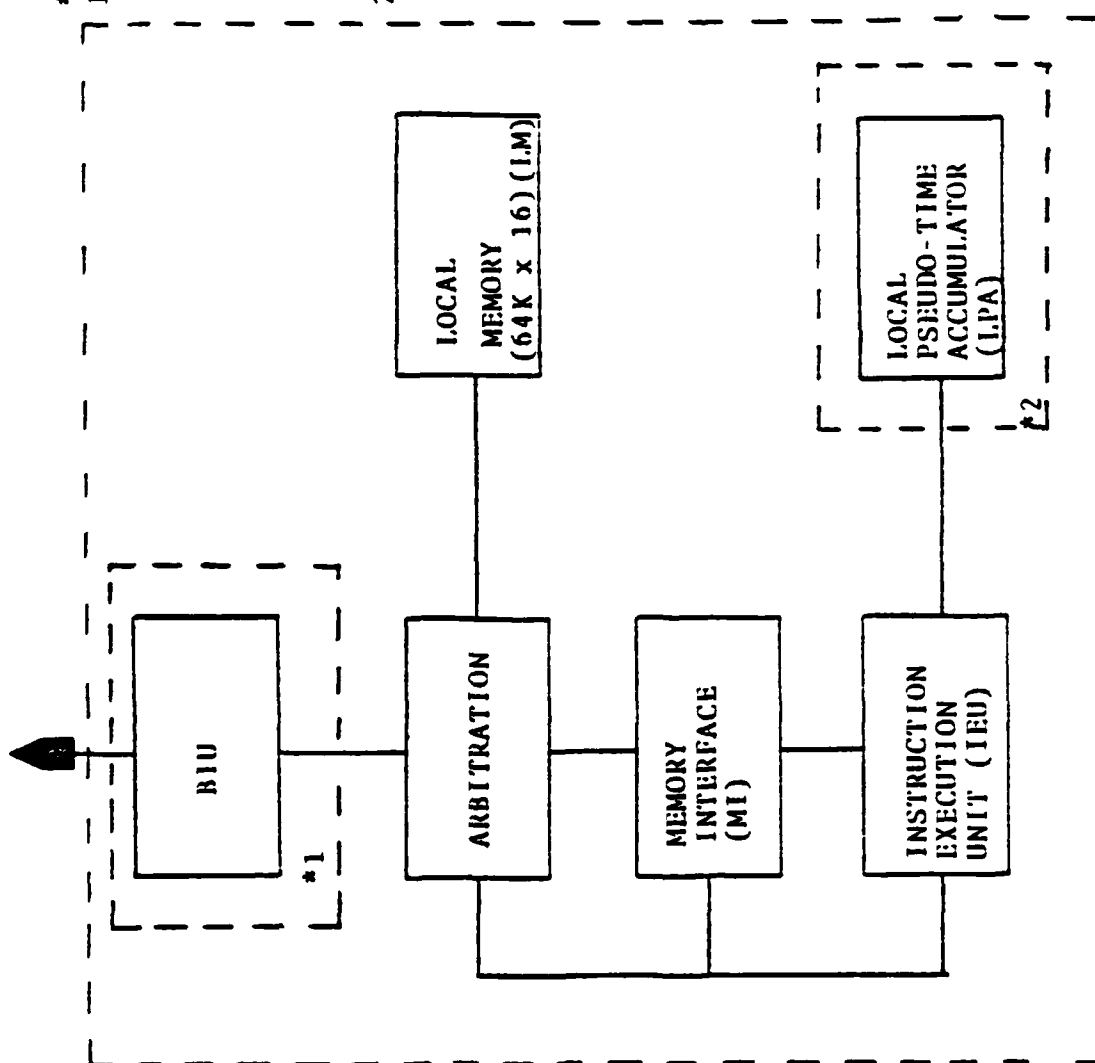


Figure 3.1.4.3 THE PROCESSING ELEMENT SEGMENT

TO THE PE SEGMENT BUS



* 1 NOTE:

The BIU is part of the Communications Bus Subsystem. Every device and interface requires a BIU.

* 2 NOTE:

The LPA is part of the Time Alignment Controller Subsystem.

Figure 3.1.4.4 THE PROCESSING ELEMENT

a. SBS side of the BIU

- 1) Transfer type - synchronous
- 2) Transfer rate - 200 NS/transfer
- 3) Voltage levels - TTL compatible
- 4) Drive capability - 133 Ω twisted pair
- 5) Device loading - 1 TTL load per bus line

b. Device side of the BIU

- 1) Transfer type - asynchronous
- 2) Transfer rate - 1 μ s
- 3) Voltage levels - TTL compatible
- 4) Drive capability - 32 TTL loads
- 5) Device loading - 2 TTL loads

3.1.5.1.2 Intersegment BIU

3.1.5.1.2.1 Functional

The Intersegment BIU shall perform all the handshaking required for the SBS protocol to interconnect a bus segment which houses PE's with the central communications segment. The characteristics for each intersegment BIU are as follows:

a. PE segment side of the Intersegment BIU

- 1) Transfer type - synchronous
- 2) Transfer rate - 200 NS
- 3) Voltage levels - TTL compatible
- 4) Drive capability - 133 Ω twisted pair
- 5) Device loading - 1 TTL load per bus line

b. Central communications side of the Intersegment BIU

- 1) Transfer type - synchronous

- 2) Transfer rate - 500 NS
- 3) Voltage levels - TTL compatible
- 4) Drive capability - 133 Ω twisted pair
- 5) Device loading - 1 TTL load per bus line

5.1.5.1.3 Facilities Control Processor Interface

3.1.5.1.3.1 Functional

The FCP Interface allows the FCP to communicate with all of the devices on SBS during system initialization. The characteristics for the FCP interface are as follows:

- a. SBS side of the FCP interface - see 3.1.5.1.1
- b. FCP side of the FCP interface - shall meet the

requirements as specified by the manufacturer of FCP.

3.1.5.1.4 Performance Monitoring System Interface

3.1.5.1.4.1 Functional

The PMS interface shall collect all the performance monitoring data output by the devices on SBS during runtime and transfers this data to the FCP. The characteristics of the PMS interface are as follows:

- a. SBS side of the PMS interface - see 3.1.5.1.1
- b. FCP side of the PMS interface - shall meet

the requirements as specified by the manufacturer of the FCP.

3.1.5.1.5 I/O Processor Interface

3.1.5.1.5.1 Functional

The IOP interface allows the FCP to control the emulated I/O data which must be supplied to the devices on SBS during runtime. The characteristics of the IOP interface are as follows:

a. SBS side of the IOP interface - see section

b. FCP side of the IOP interface - shall meet the requirements as specified by the manufacturer of the FCP.

3.1.5.2 External Subsystem Interfaces

The interconnection of MMS with external systems and/or devices requires the following interfaces:

a. ARPANET interface

b. Local terminal interfaces

3.1.5.2.1 ARPANET interface

3.1.5.2.1.1 Functional

The ARPANET interface provides remote users or systems with the capability of communications with the MMS via the FCP.

The characteristics of the interface are as follows:

a. The FCP side of the ARPANET interface - shall meet the requirements as specified by the Interface Message Processor: specification for the Interconnection of a Host on an Imp, REPT. 1822, Bolt Beranek Newman, Inc., Document Number NTIS AS-055996 (or equivalent specification).

b. ARPANET network side of the ARPANET Interface - the manufacturer of the ARPANET Interface (such as the Digital Equipment Corporations IMP-11A) shall meet the requirements as specified by the ARPANET Protocol Handbook, Document Number NTIS AD-A052594.

3.1.5.2.2 Local Terminal Interfaces

3.1.5.2.2.1 Functional

These interfaces shall allow local terminals such as CRT's, line printers, and mass storage devices to be interfaced

to the FCP. These interfaces shall meet the requirements as specified by the manufacturer of the FCP.

3.1.6 Government Furnished Property List

TBD

3.1.7 Operational and Organizational Concepts

The MMS shall satisfy the following operational and organizational concepts:

a. Ease of use for the user of the MMS shall be of high priority in the design of the MMS.

b. The user shall have the capability to communicate locally via local terminals and/or remotely via the ARPANET network with the MMS.

c. The FCP and the MMS operating systems shall provide the MMS user with the capability under software control to define the configuration to be emulated, program the MMS at all levels and load all memories, control PMS, debug, execute MMS programs, and run MMS hardware logistics.

d. The MMS shall be designed to accommodate future growth and changes without major system redesign.

e. The MMS shall provide the user with the capability to monitor the performance of a system being emulated. The user shall have access to PMS data for use in performance analysis. They shall be able to perform some PMS analysis during runtime. The use of physical probes shall not be required to measure PMS data.

f. The MMS shall contain the required hardware to perform macro and microcode debug.

g. The MMS shall provide a microcoded executive usable in each processing element to aid the user in writing the microcode emulator.

h. The MMS, when correctly configured and microcoded, shall directly execute the user's code (software/firmware) of the system being emulated.

i. The mapping of the user's code into the MMS shall be performed by the MMS.

j. The user shall have the capability to time align or not to time align the PE's and also control the resolution of the time alignment.

k. The MMS shall be designed with fault tolerant features facilitating error detection, error correction, and program restart. As a minimum, the user shall be able to perform a diagnostic test on each PE to locate faults to a functional unit. Furthermore, the MMS shall be capable of continued operation with the defective PE either removed or replaced.

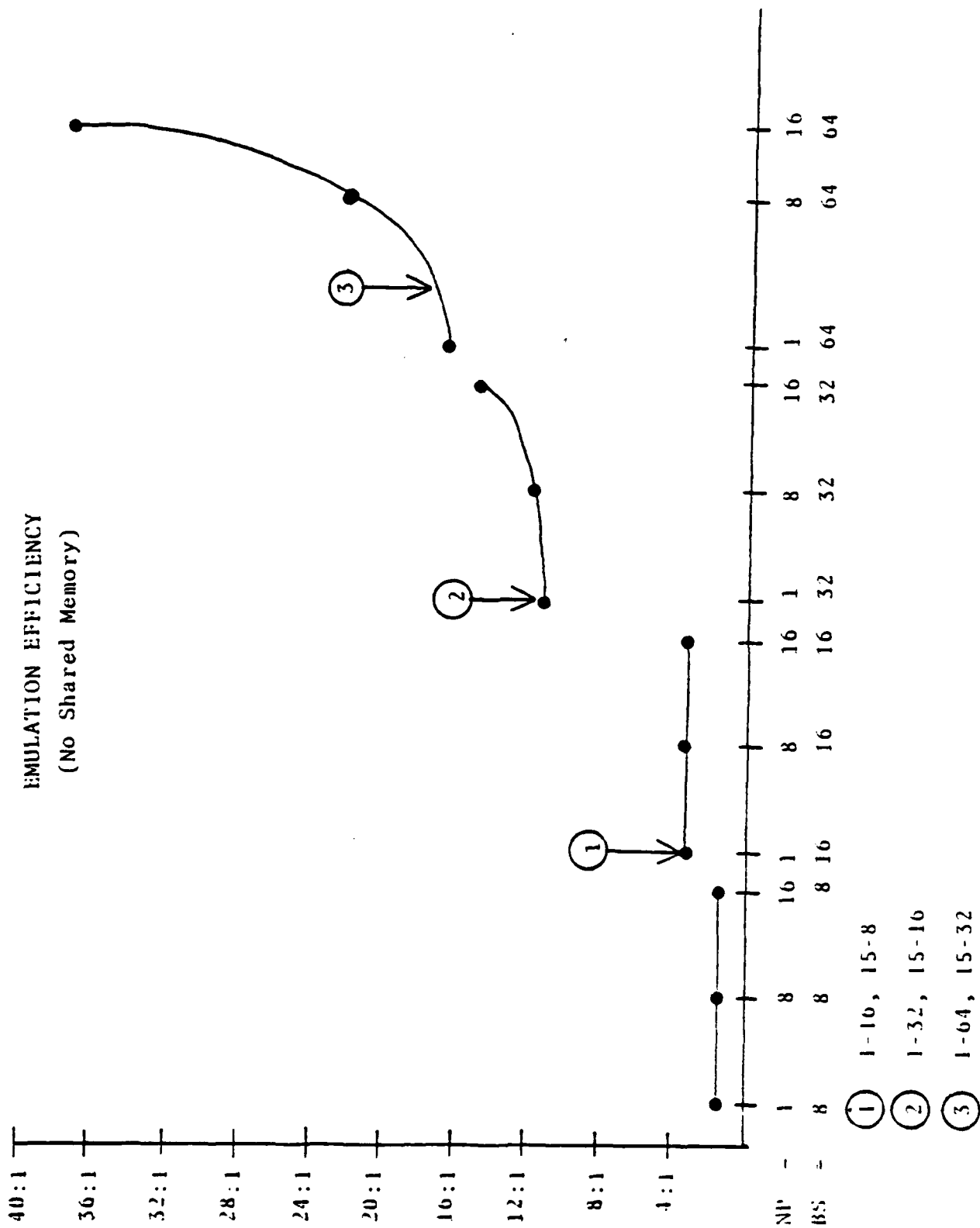
3.2 Characteristics

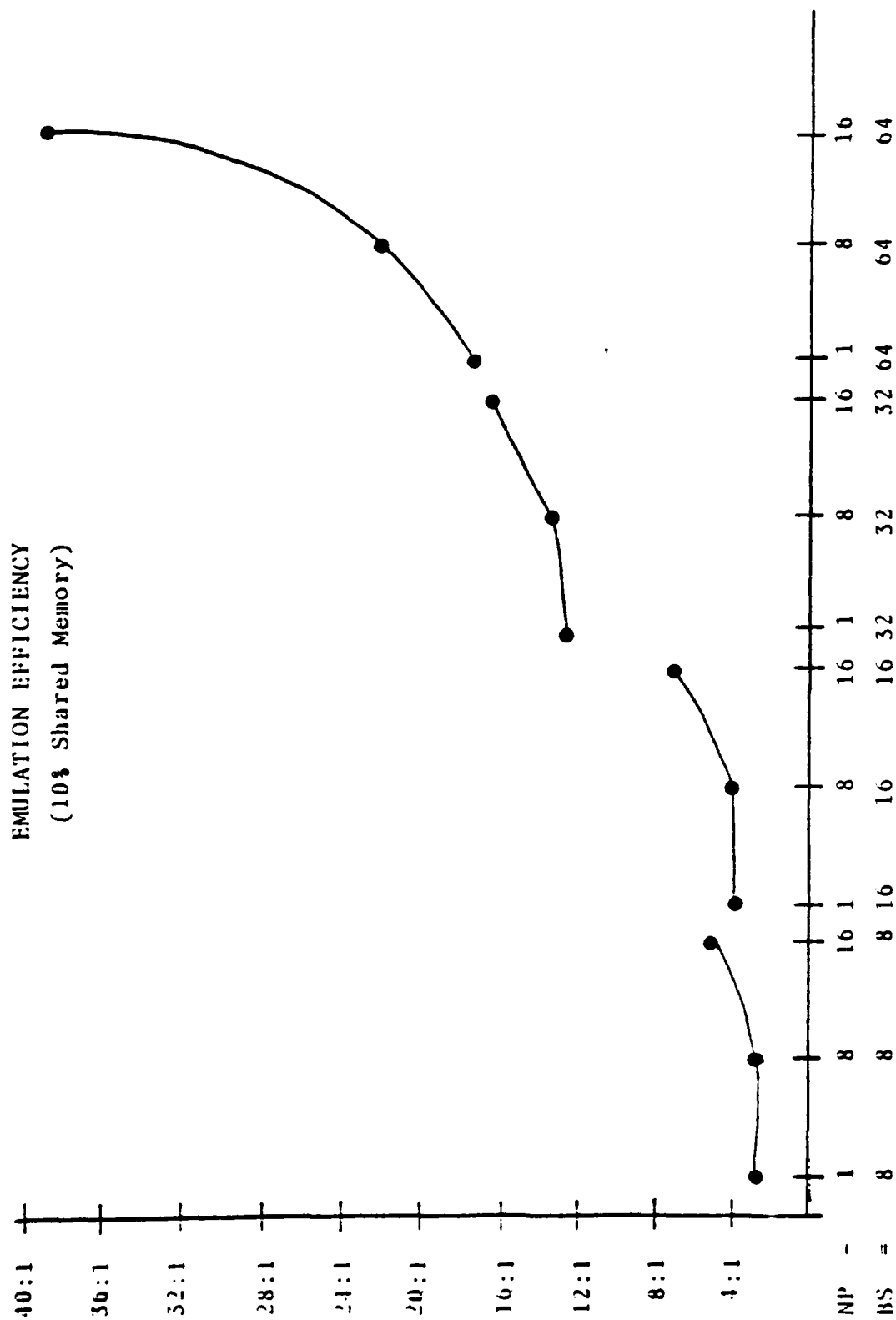
3.2.1 Performance Characteristics

The performance requirements specified in this section are based on the analysis of the MMS. This information can be found in the following document: the SAEF Final Report, Vol. I, November 1979, Contract Number F30602-78-0114, Harris Government Communications Systems Division.

3.2.1.1 Emulation Efficiency

The Emulation Efficiency (EE) for the different types of systems shall be determined from Figures 3.2.1 and 3.2.2.





TARGET SYSTEM SEGMENT

Figure 3.2.2

The different types of systems are determined by the number of processors (NP) and the processor bit size (BS). The EE's are given in terms of a slow down ratio. Typical EE's for a system with no shared memory are as follows:

a.	NP	BS	EE	4:1
	1	16		
	15	8		
b.	NP	BS	EE	12:1
	1	32		
	15	16		
c.	NP	BS	EE	17:1
	1	64		
	15	32		

The EE's for shared memory systems are given in Figure 3.2.2.

3.2.1.2 Communications Bus Subsystem

The requirements for the Communications Bus Subsystem to meet the EE's given shall be as follows:

- a. Aggregate transfer rate - 5 mega transfers/sec
- b. The average latency time for any PE to use the bus shall not exceed 1 μ s.

3.2.1.3 Emulated I/O

To meet EE's requirements, the IOP shall be required to handle all of the environmental I/O emulation. No more than 1.0% of total amount of emulated I/O shall be environmental. For a 1.0% environmental I/O the IOP shall be able to service a shared I/O request in a maximum of 0.1 MS and environmental I/O in a maximum of 1.0 MS.

3.2.1.4 Shared Resource Controller Subsystem

The Shared Resource Controller shall be able to service a shared resource request in a maximum of 3 μ S.

3.2.1.5 Time Alignment Controller Subsystem

The Time Alignment Controller Subsystem shall allow the user to control the resolution of the time alignment of PE's to within 10 NS.

3.2.1.6 PMS

When the user is operating under the PMS, the MMS shall not be required to meet the EE's given in paragraph 3.2.1.1.

3.2.2 Physical Characteristics

3.2.2.1 Physical Location

All MMS equipment shall be located at the computer facility at RADC.

3.2.2.2 Clearance Envelope

The MMS components shall be designed and installed with sufficient clearance for personnel access during maintenance and normal operation.

3.2.2.3 Mobility

All MMS racks shall be designed such that they can be relocated on a rack by rack basis within the RADC complex.

3.2.2.4 Size

The MMS, excluding the FCP, shall be housed in a maximum of 16 racks. These racks shall be standard racks, seven feet in height.

3.2.2.5 Weight

TBD

3.2.2.6 Noise Generation

TBD

3.2.2.7 Grounding

TBD

3.2.2.8 Security

TBD

3.2.3 Reliability

When operating under the environmental conditions specified in 3.2.7, the MMS shall have a mean time between failures (MTBF) of 40 hours.

3.2.4 Maintainability

Repair of failed modules shall be based on a remove-and-replace philosophy. The Mean-Time-To-Repair (MTTR) for all components within the MMS shall not exceed 2 hours.

3.2.5 Availability

The MMS shall be available to a user a minimum of 80% of the operational time.

3.2.6 System Effectiveness Models

Not applicable.

3.2.7 Environmental Conditions

The MMS shall be designed to operate in a standard commercial computer room environment.

3.2.7.1 Temperature

The mean operating temperature shall be 75°F with a deviation of $\pm 10^\circ\text{F}$.

3.2.7.2 Humidity

The mean relative humidity shall be 50% with a deviation of $\pm 5\%$.

3.2.7.3 Operating Voltage

The required operating voltage shall be 220V/110V.

3.2.7.4 Dust and Smoke

The MMS shall operate under dust and smoke conditions encountered in standard commercial computer environments.

3.2.8 Nuclear Control Requirements

Not applicable.

3.2.9 Transportability

The MMS and components thereof shall be transportable by commercial means. During transportation the MMS components shall be secured in a manner which prevents excessive shock and vibration.

3.3 Design and Construction

3.3.1 Materials, Processes and Parts

Design and construction of the MMS shall meet best commercial practices.

3.3.2 Electromagnetic Radiation

None.

3.3.3 Identification and Marking

All equipments and assemblies shall be identified with the following information as a minimum:

Manufacturer's part or model number

Manufacturer's name

Unit name

Serial number, as applicable

Revision letter, as applicable

3.3.4 Workmanship

All MMS fabricated equipment shall be constructed per the applicable engineering drawing and in accordance with Military Standard 454 Requirement 9. All equipment supplied by a subcontractor or vendor shall be constructed in accordance with workmanship criteria acceptable to RADC. "Off the shelf" hardware will, at a minimum, conform to high commercial quality standards.

3.3.5 Interchangeability

Units, assemblies and line replaceable parts shall be physically and functionally interchangeable, without modification of such items or of the equipment with which such items interface. Individual items shall not be hand picked for fit or performance.

3.3.6 Safety

The MMS design shall provide protection to personnel and equipment in accordance with the recommendations given in the Preliminary Hazard Analysis for SAEF/MMS, Document Number 6277-7751-004.

3.3.7 Human Performance/Human Engineering

Best commercial practice.

3.4 Documentation

The vendor of the MMS shall supply operation and maintenance manuals. For repair, function logistics for each unique card in the MMS shall be provided.

3.5 Logistics

3.5.1 Maintenance

The contractor shall provide 2 spare wire wrap cards for each unique wire wrap card contained in the MMS to maintain the MMS as required to meet the specifications of Section 3.2.4.

3.5.2 Supply

TBD

3.5.3 Facilities and Facility Equipment

TBD

3.6 Personnel and Training

3.6.1 Personnel

TBD

3.6.2 Training

TBD

3.7 Functional Area Characteristics

3.7.1 Communication Bus Subsystem

Provides the MMS with a high bandwidth communications path which will be an effective processor to processor, processor to control function, and processor to memory interconnect structure. This subsystem shall have a modular structure which facilitates future expansion without major system design. This subsystem shall be capable of operating with an aggregate transfer rate of 5 mega transfers/second.

3.7.1.1 SBS

Provides the Communications Bus Subsystem with a busing structure and protocol. The functional characteristics of the SBS are as follows:

a. The SBS will be a unified busing structure. This unified busing structure will be a single, wide parallel bus with a flexible information field which will handle all MMS communications.

b. All transfers over SBS will be synchronous register to register type transfers.

c. The SBS will separate device cycle time and bus cycle time. This will allow the bus to be free for use by another device while a device such as a memory is performing a read cycle.

d. The SBS will provide a minimum memory address space of 64 megabytes for storing user programs during emulation.

e. The SBS will have a segmented busing structure. Each segment will operate independently of any other segment. Intersegment communications between the bus segments which house PE's will be accomplished via central communications bus segment.

f. Each bus segment will have its own BSC.

g. Each bus segment will have the capability to be expanded to a maximum of 32 devices per segment.

h. Each device to be connected to the SBS will be connected to the SBS via a BIU.

3.7.1.2 BSC

The BSC provides for centralized bus segment control. The functional characteristics of the BSC are as follows:

a. The BSC shall provide all the required bus timing control as required by each bus segment.

b. The BSC shall provide bus segment arbitration.

3.7.1.3 BIU

The BIU provides the link between a device and the SBS. The BIU shall perform all the buffering and handshaking required by SBS to send and receive messages. The functional characteristics of the BIU are as follows:

- a. Each BIU shall have independent receive and send ports.
- b. Each BIU shall generate parity for communications to be sent.
- c. Each BIU shall detect parity errors for communications received and request for data to be resent if an error is detected.
- d. BIU's shall have a separate buffer for receiving broadcast communications.

3.7.1.4 Intersegment BIU

The Intersegment BIU shall perform in the same manner as device BIU. The Intersegment BIU shall be a modified BIU which interconnects two SBS segments instead of a device and an SBS segment. The Intersegment BIU will perform all the handshaking and buffering required by the central communications segment and a PE segment.

3.7.2 Processing Element Subsystem

Provides the MMS with all the tools necessary to handle the complete emulation of one target system processor unit exclusive of shared memory arbitration, shared I/O arbitration, environmental I/O handling, or any other function which will be centrally located in the target system. One target system (TS)

processor unit refers to any unit in a Target System which executes instructions inclusive of its memory accesses, I/O accesses to or from the unit, or internal communications associated with that unit. This subsystem shall be replicated sixty-four times to allow the emulation of a maximum of sixty-four target system processor units.

3.7.2.1 Instruction Execution Unit

Provides the Processing Element Subsystem with a structure that handles the emulation of target system processor op-codes. The functional characteristics of the IEU are as follows:

a. The IEU shall be able to be operated in a remote mode. In this mode the FCP control shall be able to download or upload any register or memory location associated with the emulation of a target system processor unit.

b. Debug features of single-step, trap or op-code, register access, or memory location, and trace of instructions shall be implemented on the macrocode level. Debug features of single-step, execute for a number of instructions, and execute for an interval of pseudo-time shall be implemented on the micro-code level. Debug of both levels will be facilitated by the use of a small firmware debug package (part of PEXE 3.7.2.1(h), which handles the interface between the emulation debug and the FCPI.

c. Performance monitoring of the target system unit shall be implemented in the IEU. The IEU shall be capable of monitoring the access of any user register, locally emulated I/O device register, or any instruction op-code. For each of the above mentioned registers or op-codes, the IEU shall be able to generate a unique message (up to 4096) and transmit this message to the FCP Subsystem. When no performance monitoring event occurs, there shall be no elapsed time in the process of determination of the possible occurrence of an event.

d. The decode of instruction op-codes shall be facilitated through the use of logic which can directly extract any one to eight bit field from a sixteen bit instruction register.

e. An arithmetic unit with 32 associated registers shall exist to perform all routine 16-bit addition, subtraction and logic functions, as well as 16-bit integer multiplication in one microcycle. The associated registers are scratch pad registers which can be logically divided into sections for the different contexts of machine use.

f. The local pseudo-time accumulator shall exist physically within the IEU, but shall be considered as a part of the Time Alignment Control Subsystem.

g. Two contexts of arithmetic condition codes shall be maintained in the PE. These shall be the microcode conditions and the TS conditions. The TS conditions shall not be required to be maintained in the same order in which they exist within the TS, but the order transformation shall not

require more than 3 microcycles when demanded. An additional sixteen bits of user defined status must be maintained such that the update or testing thereof shall be possible on one microcycle.

h. The IEU shall contain a firmware package that orders and facilitates the target system emulation package. Specifically, this firmware package, PEXE, shall provide formatting for messages outgoing from the PE, handling of PMS events, and some routing and scheduling for I/O interrupts.

i. The emulation of simple I/O devices shall be handled within the IEU. A place for storage of device registers, user written emulation code, and routing tables to reach the emulation code shall be provided. Interval timers shall be implemented and associated with each I/O action to provide the correct interval of use of interrupting devices.

j. The capability shall exist to emulate the overlap of instruction execution with memory fetches.

3.7.2.2 Local Memory

Provides the Processing Element subsystem with the storage needed for target system processor instructions. The functional characteristics of the LM are as follows:

a. The LM shall be able to handle a memory access independent of the rest of the system once a memory read or write request has been logged. For read requests, the LM shall have the capability to format a message back to the appropriate requesting device with the data included.

b. The LM shall be divided into two 64K x 8 modules so that either 8-bit or 16-bit data may be simultaneously read or written. In the case of the 8-bit data read, the data shall be right justified.

c. The LM shall be able to handle read-modify-write situations. In these situations, write requests shall be honored only from the device which made the read request.

3.7.2.3 Memory Interface

Provides the Processing Element Subsystem with the emulation of memory accesses. The functional characteristics of the MI are as follows:

a. The MI shall contain emulation tables which describe the address space of the TS processor. These tables maintain beginning and ending TS addresses and a relocation address which allows translation to an MMS address. They will describe the memory characteristics of access time, shared or local, instruction or data, and performance monitoring characteristics. Addresses generated by MI translation shall be MMS byte addresses.

b. The MI shall execute in hardware a search routine that occurs in the following manner:

- 1) examines the instruction/data characteristic provided with the address to be translated,
- 2) initially compares the provided address to the descriptor block last matched in a search that has the same instruction/data characteristic

3) searches all descriptors with the same characteristics.

4) will then search the remaining descriptors if specified by the emulation.

c. The MI translation tables shall be downloaded in the same manner as registers and memory in the IEU through the remote mode and the internal data bus.

d. The MI shall operate independently from the IEU in the pursuit of memory accesses once the command has been given to translate an address. The MI shall have its own set of I/O ports into the BIU of the PE. In case of the emulation of 32 or 64 bit machines, the MI shall act to make the multiple accesses necessary without special instruction by the IEU. The IEU shall provide a signal to indicate when ready for next 16-bit data element for read operations and a signal to indicate when the next 16-bit element is available for write operations.

e. The MI shall provide a byte/word signal on the bus with each memory access. This signal will indicate to the LM module that either an 8 bit or 16 bit data element is the object.

f. The MI shall detect which blocks of TS address space are to be monitored by the PMS.

g. The MI shall detect the need for making a request to the shared resource controller for the memory element accessed. It shall distinguish between elements shared over a block of time or on a cycle-by-cycle basis. In the former case, the MI shall present this information to the IEU and discontinue the memory access until ordered to complete it. In the latter case, the MI

shall communicate with and receive access permission from the shared resource controller. Then it shall continue the memory access. The request/grant sequence shall be performed exactly once per logical TS memory access.

h. The MI shall communicate directly to the LPA for the purpose of updating pseudo-time due to a memory access.

i. The MI shall detect all memory mapped I/O, discontinue any memory access action, and inform the IEU that I/O is to be performed.

3.7.2.4 Message Arbitration Logic

Provides the Processing Element Subsystem with the message switching necessary between the BIU, the LM, the IEU, and the MI.

3.7.3 Broadcast Communications Controller Subsystem

Provides the MMS with an effective method of emulating SIMD and MISD machines. The Broadcast Communications Controller Subsystem shall provide the following modes:

Mode 1: Any PE shall be able to broadcast to all of the PE's collectively on the same segment via the segment broadcast controller.

Mode 2: Any PE shall be able to broadcast to all of the PE's collectively on any other segment via the broadcast controller of the receiving segment.

Mode 3: Any PE shall be able to broadcast to all of the PE's collectively via the master broadcast controller.

3.7.3.1 Segment Broadcast Controller

Provides the Broadcast Communications Subsystem with broadcast control for each segment. The functional characteristics are as follows:

- a. A PE needing to send a broadcast communication shall first send the broadcast communication to the proper segment Broadcast Controller.

- b. The Broadcast Controller receiving a broadcast communication shall send the broadcast communication only when all PE's are ready to accept the broadcast communication.

- c. A Segment Broadcast Controller shall be able to detect a hardware failure if any PE on the controller's segment fails to accept a broadcast communication.

3.7.3.2 Master Broadcast Controller

Provides the Broadcast Communications Control Subsystem with the capability to broadcast to all bus segments collectively. The functional characteristics are as follows:

- a. When the Master Broadcast Controller receives a broadcast communication, it shall send the broadcast communication only when all Intersegment BIU's are ready to accept the broadcast communications.

- b. The Intersegment BIU's send the broadcast communications to the segment broadcast controllers.

3.7.4 Time Alignment Controller Subsystem

Provides the MMS with the capability of controlling the emulation process with respect to the actual run-time of the system being emulated. This emulation control-time will be referred to as Pseudo-Time (PT) hereinafter.

3.7.4.1 TAC

Provides the MMS with a means of advancing or stopping PT. The TAC shall provide the following modes of operation:

Mode 1: Free run at maximum rate - continuously advance PT at a constant rate unless a stop is sent by an LPA.

Mode 2: Free run at slow rate - continuously advance PT at a constant rate unless a stop is sent by an LPA.

Mode 3: Burst Mode - Advance PT by the amount specified by the user and stop.

Mode 4: Single Step - Advance PT by a single increment and stop.

3.7.4.2 LPA

Provides in hardware the time alignment function for each PE. The function characteristics of the LPA are as follows:

a. Each Processing Element Subsystem shall contain an LPA.

b. The LPA shall perform a comparison of the amount of PT issued by the TAC and amount of PT used by the PE for instruction execution.

c. If the PE is behind the TAC in PT the LPA will send a stop message to the TAC.

d. If the PE is ahead of the TAC in PT the LPA will send an idle message to the PE

e. The user shall be able to program the resolution of how closely a PE must be aligned with PT. The LPA shall provide the user with a maximum resolution of 10 NS and a minimum resolution of 640 μ S.

f. The LPA shall be able to stop the TAC within 1 μ S after detecting the PE is behind the TAC in PT.

3.7.5 Emulated Shared Resource Controller Subsystem

The Emulated Shared Resource Controller Subsystem shall contain additional software as required to control emulated shared resources.

3.7.6 Facilities Control Processor Subsystem

Provides the functions of systems initialization, collection of PMS data, emulated I/O control, and interfacing the MMS to the external world.

3.7.6.1 FCP

Provides the central control function for the MMS. The entire MMS shall be under FCP software control.

3.7.6.2 FCP Interface

Provides the following functions during system initialization and debug:

a. Down load blocks of data - by loading the FCP interface with an MMS starting memory address, a FCP starting memory address, and a transfer count, the FCP interface shall move the contents of a block of FCP memory to a specified MMS memory block.

b. Up-loading blocks of data - by loading the FCP interface with an MMS starting memory address, a FCP starting memory address, and a block count, the FCP interface shall move the contents of a specified block of MMS memory to the FCP.

c. Verify block of data - by loading the FCP interface with an MMS starting memory address, a FCP starting memory address, and a block count, the FCP interface shall compare the contents of the two memory blocks.

3.7.6.3 PMS Interface

The functions of PMS shall be divided into hardware functions and software functions. The PMS functions performed in hardware are implemented in the PMS interface (PMSI). This interface shall be required to perform the following PMS functions:

a. DMA PMS data to the FCP processor's memory - PMSI shall DMA all PMS data to two buffer areas in the FCP memory. When one buffer is full, PMSI interrupts the FCP and moves the full buffer to off-line storage and at the same time begins filling the second buffer with data.

b. Detect real time events - The PMSI shall store a set of events the user desires to monitor during run time. All incoming PMS data is compared against these real time events. When a real time event is detected, PMSI interrupts the FCP which reads the data from the interface.

c. Stop master pseudo-time - when PMS data is being sent faster than the interface can service the bus, master pseudo-time shall be halted until the pending PMS events can be serviced.

d. Detect the start of an event or a periodic event - PMSI shall provide a programmable event time which the user can use to inform the FCP some type of action needs to be taken.

e. Insertion of time tag to PMS data - PMSI shall insert a PMS time tag to incoming PMS data.

3.7.6.4 IOP Interface

Provides an interface which performs the I/O emulation tasks associated with communications between the FCP and the PE. There shall be three main components within the IOP interface:

- a. IOP Ports,
- b. Interprocessor Communications Buffer (IPC),
- c. Interval Timers.

3.7.6.4.1 IOP Ports

Provides a PE with set of port from which the PE can request I/O data.

3.7.6.4.2 IPC Buffer

Provides the FCP with a port from which the FCP can send I/O data to a PE in the form of an IPC.

3.7.6.4.3 Interval Timers

A bank of timers with Pseudo-Time as an input shall be provided so that the IOP can effectively emulate I/O devices or I/O data streams with proper Pseudo-Time synchronization.

3.7.6.5 ARPANET Interface

Provides the user with the capability to remotely use the MMS.

3.7.6.6 Local Terminal Interfaces

Provides for the connection of CRT's, printers, and mass storage devices to the FCP.

4.0 QUALITY ASSURANCE PROVISIONS

4.1 General

The equipment supplier shall implement and maintain a documented Quality Assurance System, which assures supplies conform to the requirements of this specification, and which is acceptable to RADC. The Quality System, procedures and sufficient criteria shall be utilized to assure that products are adequately inspected, evaluated and tested. Except as otherwise specified, the supplier may utilize his own facilities or any facility acceptable to RADC.

Implementation of the Quality System shall be based upon consideration of the technical and manufacturing aspects of production and related engineering design and materials. Adequate quality shall be assured throughout all areas of performance; for example, design, development, fabrication, processing, assembly, inspection, test, packaging, shipping, etc. The system shall provide for the prevention and read detection of discrepancies and for timely and positive corrective action.

4.1.1 Responsibility for Inspection

Unless otherwise specified in the contract or order, the supplier is responsible for the performance of all inspection requirements as outlined here. The supplier's plans, procedures, criteria and results will be continually assessed to assure supplies and/or services conform to specification requirements. Items produced to this specification may be subject to source

inspection by authorized representatives of RADC. The correction of deficiencies shall be the responsibility of the supplier, and shall be subject to RADC concurrence.

The actions of, or inspection by, RADC representatives shall in no way relieve the supplier of full responsibility of designing, fabricating and assembling the equipment, and providing adequate packaging for shipment to ensure that, after shipment and installation, the equipment will operate in full compliance with this specification.

4.2 Quality Conformance Inspection

TBD

5.0 PREPARATION FOR DELIVERY

TBD

6.0 APPENDIX

6.1 Appendix A

A glossary of acronyms and abbreviations follows:

BS	Bit Size
BIU	Bus Interface Unit
BSC	Bus Segment Controller
CPU	Central Processing Unit
EE	Emulation Efficiency
FCP	Facilities Control Processor
IEU	Instruction Execution Unit
I/O	Input/Output
IOP	Input/Output Processor
IPC	Interprocessor Communications
LM	Local Memory
LPA	Local Pseudo-Time Accumulator
MI	Memory Interface
MISD	Multiple Instruction Single Data
MMS	Multimicroprocessor System
MPT	Master Pseudo-Time
NP	Number of Processor
PE	Processing Element
PEXE	Processing Element Executive
PMS	Performance Monitor System
PMSI	Performance Monitoring System Interface
PT	Pseudo-Time
RADC	Rome Air Development Center

SAEF	System Architecture Evaluation Facility
SBS	Synchronous Busing Structure
SIMD	Single Instruction Multiple Data
SRC	Shared Resource Controller
TAC	Time Alignment Controller
TS	Target System

APPENDIX K

The MMS Software Specifications

TABLE OF CONTENTS

<u>Paragraph</u>	<u>Title</u>	<u>Page</u>
1.0	SCOPE -----	K-1
1.1	Identification -----	K-1
1.2	Functional Summary -----	K-1
2.0	APPLICABLE DOCUMENTS-----	K-2
3.0	REQUIREMENTS-----	K-2
3.1	Computer Program Definition-----	K-3
3.2	Detailed Functional Requirements -----	K-7
3.2.1	MMS Diagnostic Programs (Module A0) -----	K-7
3.2.1.1	Inputs -----	K-7
3.2.1.2	Processing -----	K-12
3.2.1.3	Outputs -----	K-12
3.2.2	Microcode Assembler (Module C0) -----	K-12
3.2.2.1	Inputs -----	K-12
3.2.2.1.1	Source Program -----	K-12
3.2.2.1.2	Encoder Table -----	K-14
3.2.2.1.3	Error Table -----	K-14
3.2.2.2	Processing -----	K-14
3.2.2.3	Outputs -----	K-14
3.2.3	PE Executive or PEXE (Module D0) -----	K-15
3.2.3.1	Inputs -----	K-15
3.2.3.2	Processing -----	K-15
3.2.3.2.1	Initialization of Data -----	K-15
3.2.3.2.2	DEBUG Microcode -----	K-15
3.2.3.2.3	DEBUG/PMS Interrupt -----	K-15
3.2.3.2.4	Error Interrupt -----	K-17
3.2.3.2.5	H/W Interval Timer Interrupt -----	K-18
3.2.3.2.6	I/O Request -----	K-19
3.2.3.2.7	Scheduler -----	K-19
3.2.3.2.8	Pseudo-Clock Alignment -----	K-19
3.2.3.2.9	Clean Up -----	K-20
3.2.3.3	Outputs -----	K-20
3.2.4	System Generation Loader or SGL (Module E0)-----	K-20
3.2.4.1	Inputs -----	K-20
3.2.4.2	Processing -----	K-22
3.2.4.3	Outputs -----	K-22
3.2.5	Run-Time Executive or RTE (Module F0)-----	K-23
3.2.5.1	Inputs -----	K-23
3.2.5.2	Processing -----	K-23
3.2.5.3	Outputs -----	K-25

TABLE OF CONTENTS (Cont'd)

<u>Paragraph</u>	<u>Title</u>	<u>Page</u>
3.2.6	Post Mortem Dump or PMD (Module GØ)-----	K-25
3.2.6.1	Inputs -----	K-25
3.2.6.2	Processing -----	K-25
3.2.6.3	Outputs -----	K-27
3.2.7	PMS Off-Line Analysis or PMS (Module HØ)-----	K-27
3.2.7.1	Inputs -----	K-27
3.2.7.2	Processing -----	K-29
3.2.7.3	Outputs -----	K-30
3.2.8	Target System Emulation or TS (Module IØ)-----	K-30
3.2.8.1	Inputs -----	K-30
3.2.8.2	Processing -----	K-32
3.2.8.3	Outputs -----	K-32
3.2.9	Executive for IOP (IOPE)-----	K-33
3.2.9.1	Inputs -----	K-33
3.2.9.2	Processing -----	K-33
3.2.9.3	Outputs -----	K-35
3.2.10	Executive for PMSI (PMSIE) -----	K-35
3.2.10.1	Inputs -----	K-35
3.2.10.2	Processing -----	K-35
3.2.10.3	Outputs -----	K-37
3.2.11	Executive for SRC (SRCE) -----	K-37
3.2.11.1	Inputs -----	K-37
3.2.11.2	Processing -----	K-37
3.2.11.3	Outputs -----	K-39
3.3	Adaption -----	K-39
4.0	QUALITY ASSURANCE PROVISION -----	K-39
4.1	Introduction -----	K-39
4.1.1	Computer Subprogram Testing -----	K-39
4.1.2	Computer Program Testing -----	K-39
4.1.3	Computer Program Acceptance Testing -----	K-40
5.0	PREPARATION FOR DELIVERY -----	K-40
6.0	NOTES -----	K-40
10.0	APPENDIX I -----	K-40

1.0 SCOPE

1.1 Identification

This specification establishes the requirements for performance, design, test and qualification of the software for the Multiple Microprocessor System (MMS). The software effort shall be hereinafter referred to as Software or S/W.

1.2 Functional Summary

The primary purpose of this specification is to provide the detailed requirements against which the design, coding, checkout, test, and qualification of the S/W proceeds in the development of computer programs to support the MMS configuration of a Facility Control Processor networked with the desired number of microprocessors, known as PE's (Physical Elements).

a. The purpose of MMS is to utilize this configuration structure to emulate a wide range of target multiprocessor systems. These systems are to be either single - or multiple-instruction with either single - or multiple-data (SISD, SIMD, MISD, MIMD).

b. MMS will consist of one commercially available mini-computer, several specially - dedicated PE's, and 16-64 PE's available for user applications.

c. The specially - dedicated PE's will accomplish their functions either completely on the hardware level or by firmware. These executives which become firmware will be part of the S/W design and programming effort.

d. The PE's available for the user will have a replicated executive, called PEXE.

e. S/W will be necessary to enable the design and development of the emulation modules which are configured together to produce the desired Target System (TS).

f. S/W will be necessary to enable the user to select from the library of emulated modules, and then to do the configuration into the TS.

g. S/W will be necessary to enable the user to configure into the TS his operating system, his application programs, and the environment of the TS. The environment consists of the Input/Output (I/O) to the TS.

h. S/W will be necessary to enable the user to execute the TS. The modes of execution are straight execution, DEBUG, and PMS. The user has tools to help evaluate the design or the performance of the TS.

i. S/W will be necessary to perform other specified functions to allow additional flexibility to the MMS user.

2.0 APPLICABLE DOCUMENTS

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referred herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

- a. NTIS AD-A052594 ARPANET Protocol Handbook
- b. Contract F30602-78-0114 Final Report Vol. I, SAEF Hardware Package, Nov 1979
- c. Contract F30602-78-0114 Final Report Vol. II, SAEF Software Package, Nov. 1979
- d. Contract F30602-78-0114 Appendix J, "The MMS Hardware Specifications"

3.0 REQUIREMENTS

3.1 Computer Program Definition

The S/W required for the MMS configuration will allow Rome Air Development Center (RADC) to emulate and to execute an open-ended variety of TS's. These TS's shall include a single computer system or a network of distributed systems under any design philosophy. The architecture of the TS's shall be one of four possible types: either single- or multiple-instruction with either single - or multiple-date (SISD, SIMD, MISD, MIMD).

a. Since MMS is to be incorporated into an existing facility at RADC, called the System Architecture Evaluation Facility (SAEF), the S/W shall be compatible and callable from the other systems of SAEF.

b. Since MMS is to be available to remote users, the S/W of MMS shall be usable from ARPANET.

c. To these requirements, the MMS S/W shall consist of eleven modules:

1. MMS Diagnostic Programs
2. Microcode Assembler
3. PE Executive (PEXE)
4. System Generation Loader (SGL)
5. Run-Time Executive (RTE)
6. Post Mortem Dump (PMD)
7. PMS Off-Line Analysis (PMS)
8. Target System
9. Executive for I/O Processor (IOP)
10. Executive for PMS Interface (PMSI)
11. Executive for Shared Resource Controller (SRC)

Figure 3.1-c illustrates the utilization of most of these modules. A programmer who understands the hardware to be emulated for a TS will generate the required libraries by utilizing the Microcode Assembler. PEXE is also available on disk from the Facility Control Processor (FCP, which is a commercially obtainable mini-computer/midicomputer). When the general user wants to configure his TS computer network, he utilizes SGL. At initialization of SGL, the diagnostic programs are loaded into FCP and into all PE's, including the several special-purpose PE's. Next within SGL, the user configures the TS by utilizing the library of emulated modules. The user will utilize RTE to load the applications programs with operating systems into every emulated computer system, to configure all TS I/O, and then to execute the TS under one of three modes: straight execution, execution under DEBUG, or execution under PMS. If the user desires, at termination of any execution mode, he may exercise the Post Mortem Dump program. If he has collected PMS data for off-line analysis, he may massage the PMS data file through the PMS Off-line Analysis program.

d. Several other S/W packages are included in the previous paragraph (3.1-c). One is a sample TS, which consists of an emulated PDP-11/45 Computer System on a sample bus (Bus G) with two emulated INTEL 8080's. The 8080's share a memory device, while the PDP-11/45 has mag tape for input and a disk for output (see Figure 3.1-d). The other three S/W packages are Executives for three of the special-purpose PE's: the IOP, the PMSI, and the SRC; these will become firmware within their respective PE's.

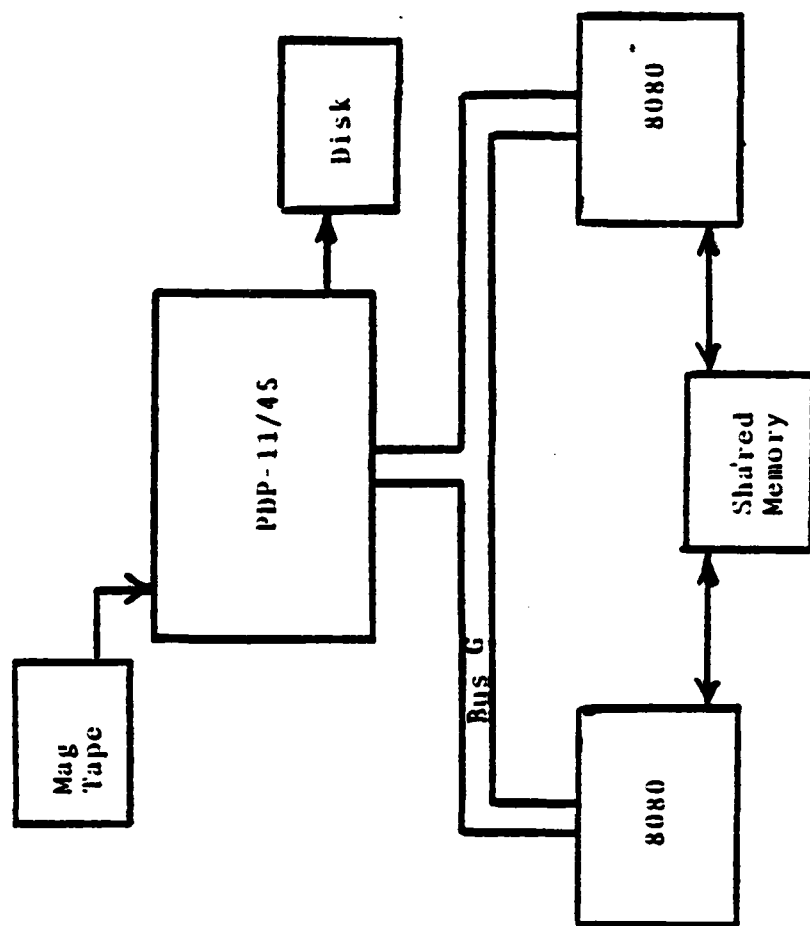


Figure 3.1d TS To Be Emulated

3.2 Detailed Functional Requirements

3.2.1 MMS Diagnostic Programs (Module A0)

As illustrated in Figure 3.2.1a-d, A0 consists of three parts: the Master for FCP, Sub-Masters within each PE, and Slaves within the "Big 7", which are the special-purpose PE's. This package A0 shall be a stand-alone program to be used for hardware error-checking; or its modified version shall be callable by SGL to ascertain the operation of all PE's, even the special-purpose PE's. Therefore, the preparation for running diagnostics shall be the loading of the Master into the FCP, the downloading of a copy of the Sub-Master to all general PE's, and the downloading of the individual Slaves to the necessary special-purpose PE's (Big 7").

a. NOTE: The term "Big 7" refers to the seven special-purpose PE's on the Master Segment. These are described within documents b,c (Section 10), d of paragraph "2.0 APPLICABLE DOCUMENTS". As a review, these are:

1. FCPI - Facility Control Processor Interface
2. PMSI - Performance Monitoring System Interface
3. IOP - Input/Output Processor
4. MBC - Master Broadcast Controller
5. SRC - Shared Resource Controller
6. S5BC - Segment 5 Bus Controller
7. TAC - Time Alignment Controller

3.2.1.1 Inputs

To initiate execution of the Diagnostic Programs, the user will wait for all Sub-Masters and Slaves to be downloaded. The Master will interrogate the user as to which diagnostic tests he wishes to process. These shall be itemized on user CRT within

AB
("Big 7")

AB
(Pi)

AB
(FCP)

Figure 3.2.1a Top-Down Design Chart

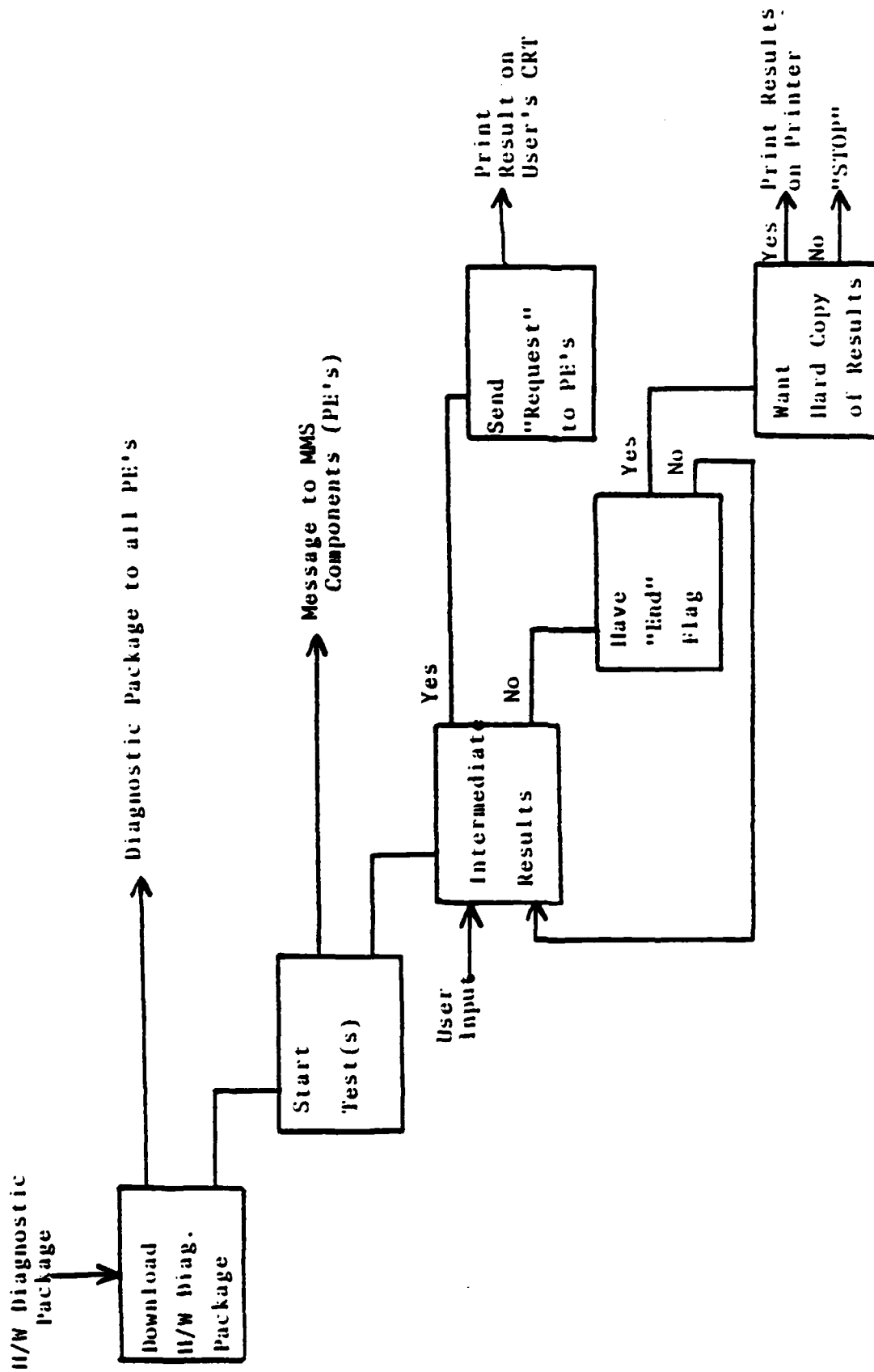


Figure 3.2.1b

AØ (RCP)

U/W Diagnostics

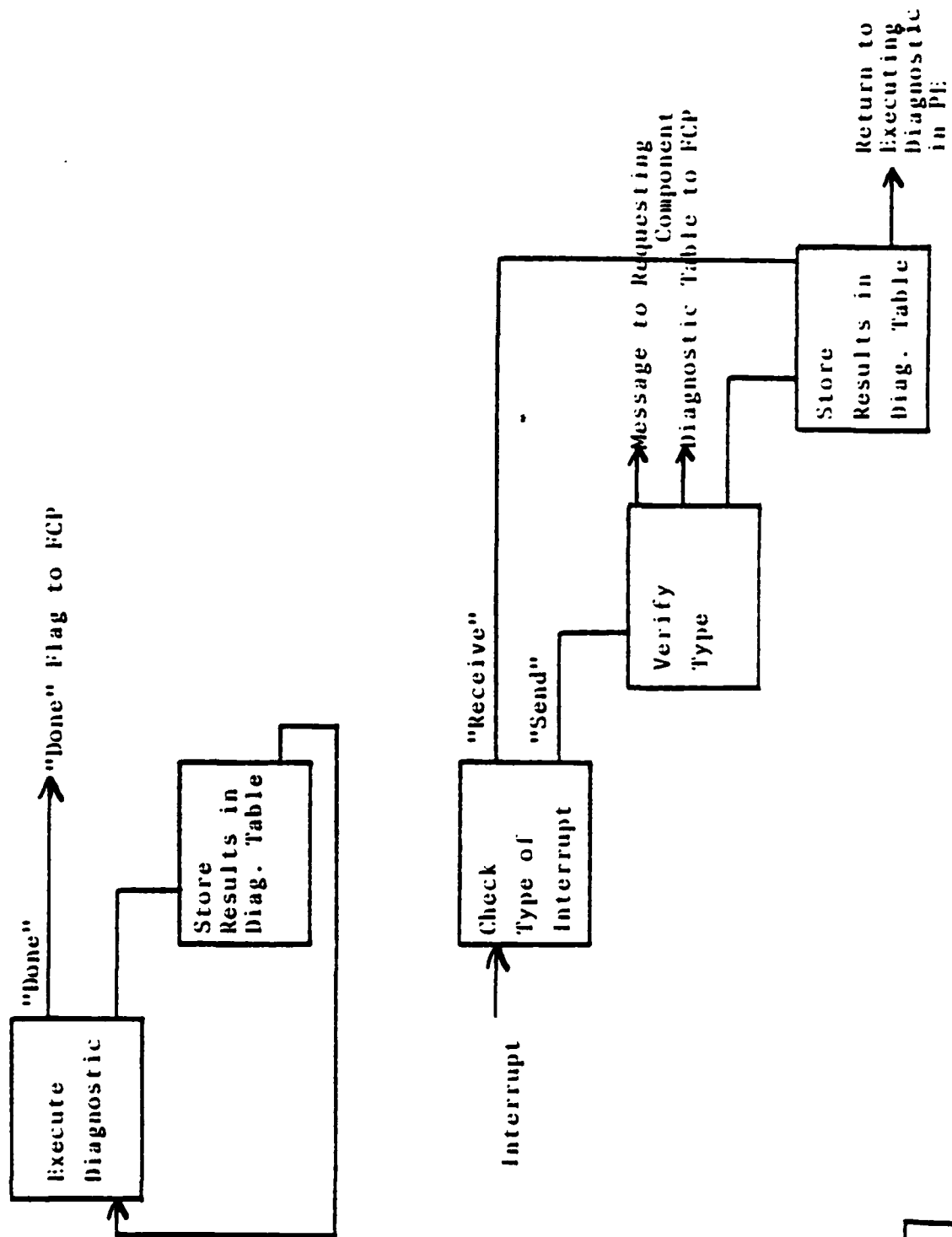


Figure 3.2.1c

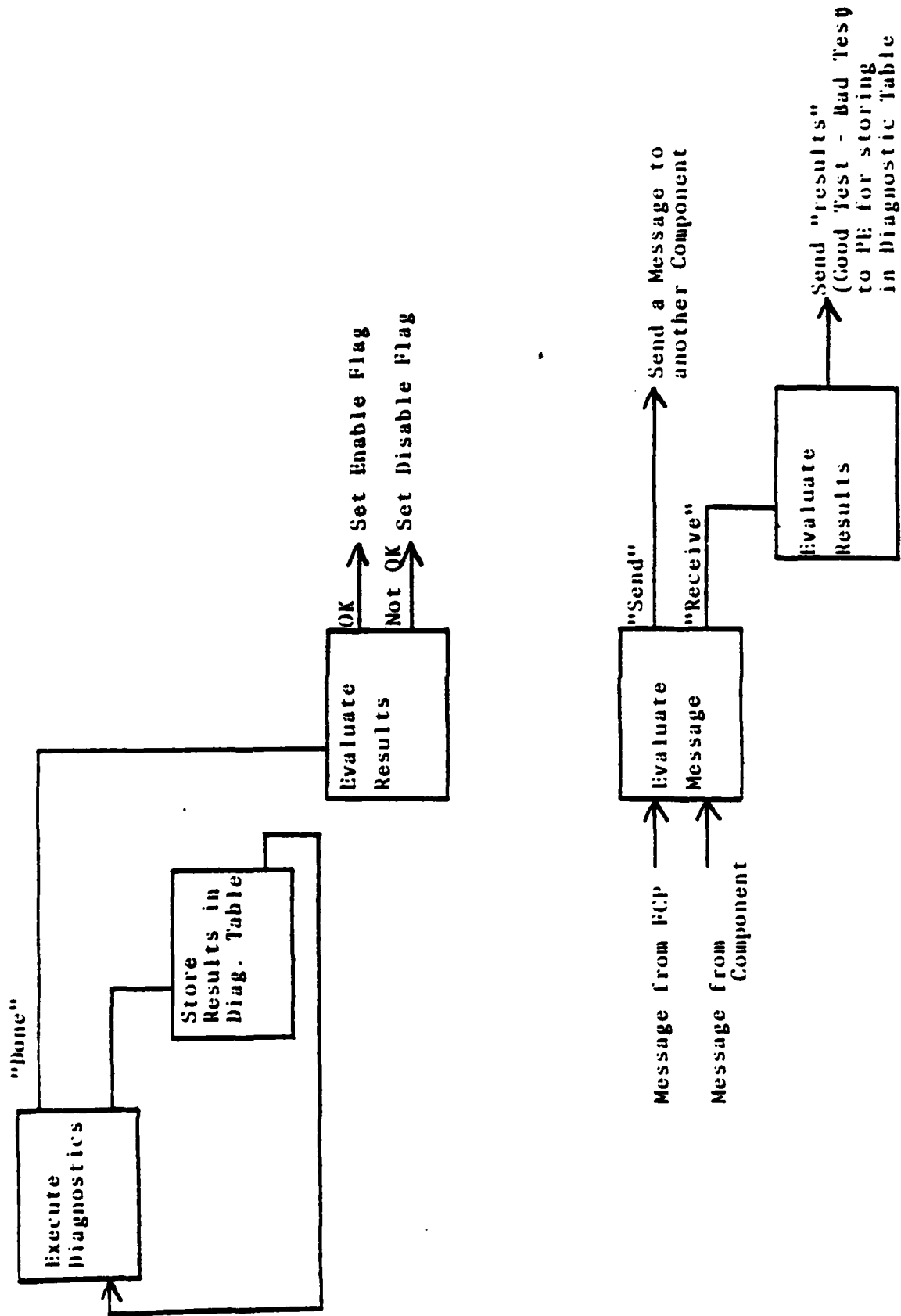


Figure 3.2.1d

AØ ("Big 7")
w/w Diagnostics

a menu. Thus, selected PE's can be tested, or all tests can be selected. Once the user indicates start, all PE's shall be synchronized together, or executed in lockstep.

3.2.1.2 Processing

Once the Master within FCP sends a start signal to TAC, it shall start sending pseudo-clock ticks to all PE's. Thus, the Sub-Masters execute synchronously, or in lockstep. The Sub-Masters shall process tests to check out all H/W within its respective PE. Then selected PE's shall message the H/W functions of the "Big 7". Each Sub-Master shall collect data relative to all tests processed by it. When each Sub-Master finishes, it shall send a message to the Master. Then the Master shall tell the individual Sub-Master to upload its table of results.

3.2.1.3 Outputs

The Master shall evaluate the results and shall then print:

- a. a detailed report of all tests processed and the results.
- b. a summary report indicating which PE's and which special-purpose PE's are operational, i.e., fully operational or down.

3.2.2 Microcode Assembler (Module C0)

The Microcode Assembler is illustrated in Figure 3.2.2.

3.2.2.1 Inputs

3.2.2.1.1 Source Program

The user, who is usually one knowledgeable of hardware characteristics, will input a source program written in the specified Microcode format. The Source Program will be emulating a Computer System instruction set or will be emulating an I/O processor.

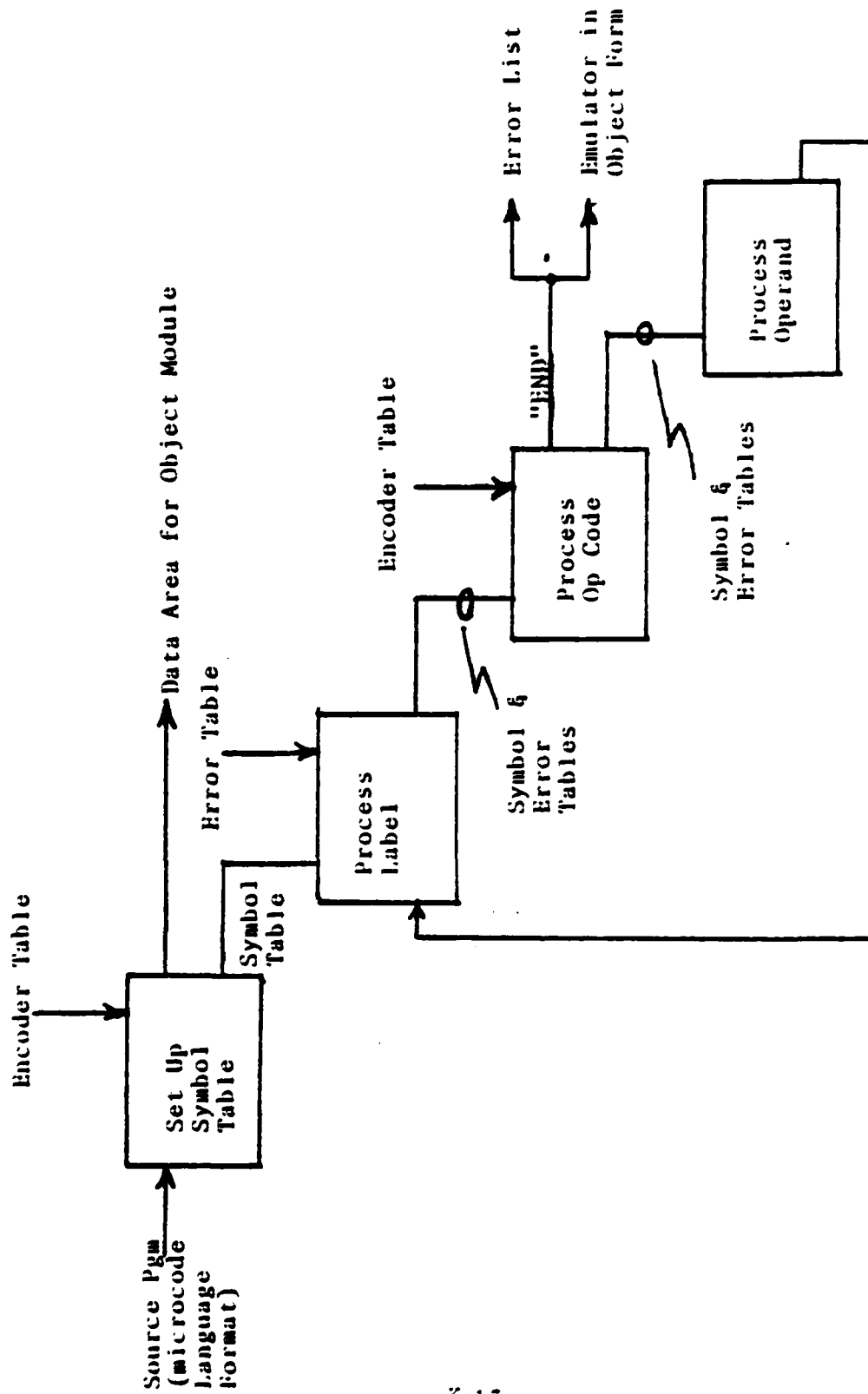


Figure 3.2.2

Microcode
Assembler

CØ

3.2.2.1.2 Encoder Table

The Encoder Table shall contain all instructions the Microcode Assembler and the comparable format for formulating the object form of the instructions.

3.2.2.1.3 Error Table

The Error Table shall contain a numeric value to correspond to all error conditions found when processing the Microcode Assembler. This table shall be built during execution of the Microcode Assembler.

3.2.2.2 Processing

As the Source Program is read, the Data Area shall be created. Also, the Symbol Table shall be generated. During pass two, the Symbol Table shall be utilized to verify correctness of the label of the instruction; if the PC (Program Counter) does not correspond, the Error Table shall reflect this problem. Then, utilizing the Encoder Table and the Symbol Table, the OpCode field of the instruction shall be processed to build the corresponding object form of the OpCode. If any errors were found, the Error Table shall be updated. Then the Operand field shall be processed to complete the development of the object form of the instruction; the Symbol Table shall be utilized to select type of the Operand and its PC value. Again the Error Table shall be updated to reflect any error conditions. All instructions shall be processed individually until the "END-type" instruction is found.

3.2.2.3 Outputs

If the Error Table contains any entries, the Error Table shall be available to the user as line printer output and/or the CRT display. If the Error Table is empty, then the Emulator in object form is ready. The Microcode Assembler shall store this

object routine within the proper library of emulators.

3.2.3 PE Executive or PEXE (Module D0)

PEXE is illustrated in Figure 3.2.3. PEXE shall consist of nine phases: Initialization and eight interrupt-driven phases.

3.2.3.1 Inputs

The inputs to PEXE shall be from the RTE when the user is ready to execute his configured TS or shall be from one of the user-designed emulation modules.

3.2.3.2 Processing

3.2.3.2.1 Initialization of Data

This phase shall initialize the data area for PEXE and for all emulator modules of the TS stored within that PE. These emulator modules include the emulated instruction set and all emulated I/O processes. Then PEXE shall send a "ready" message to the user; at this time the user is executing the RTE package.

3.2.3.2.2 DEBUG Microcode

This phase shall contain the processes necessary to interact with the hardware to allow DEBUG on the microcode level. The user shall be able to set and clear breakpoints, to set traces based upon microinstruction types, upon n number of microinstructions, upon length of pseudo-time, or upon various dump and load status commands.

3.2.3.2.3 DEBUG/PMS Interrupt

Whenever a macro-level program, I/O process, memory access operation, etc., has ascertained that the DEBUG/PMS flag has been enabled for that operation, that operation will send an interrupt to PEXE. PEXE will then handle the data collection process through PMSI to the desired location in FCP memory.

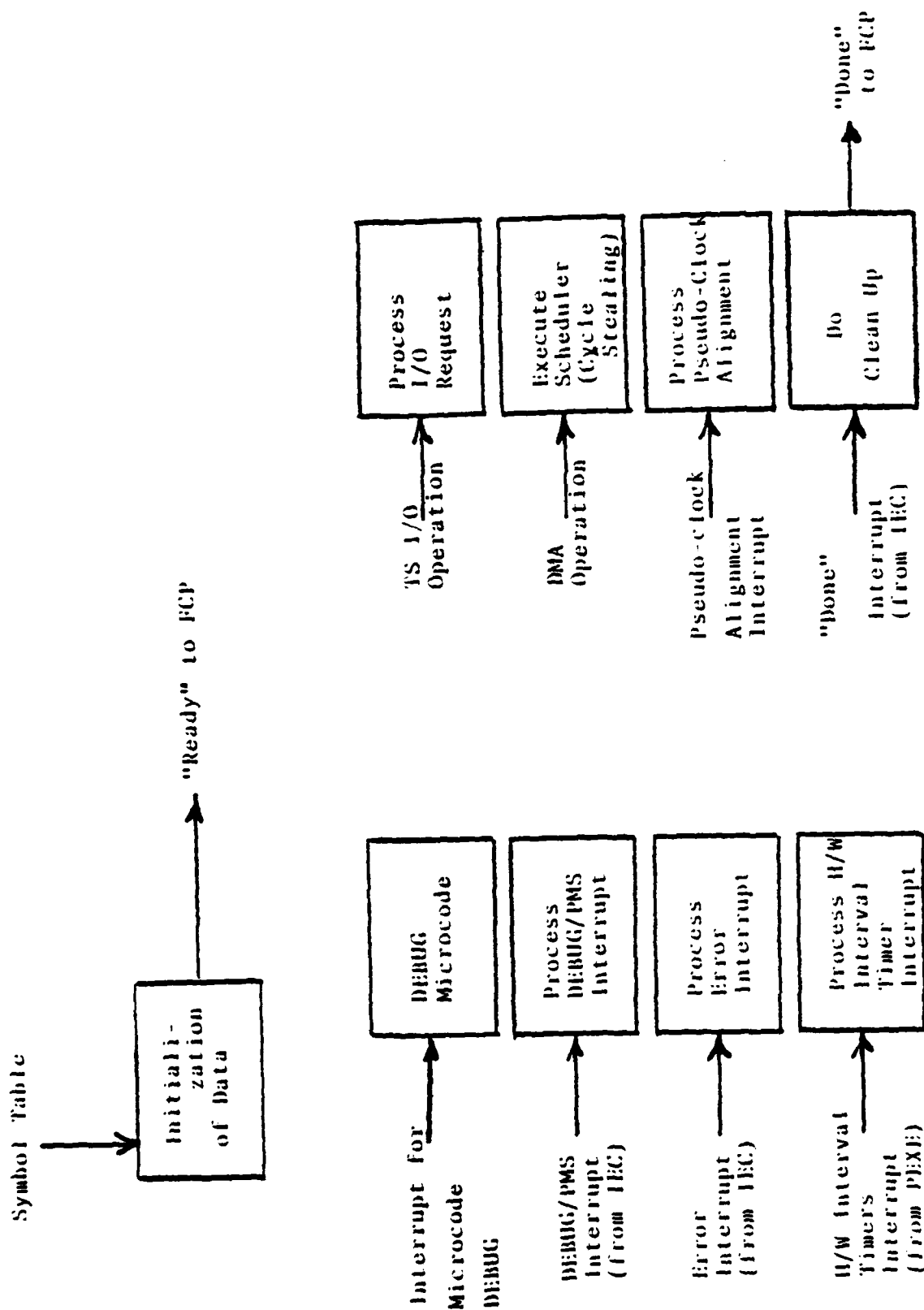


Figure 3.2.3

DØ

PEXE

Upon entering PEXE, because of the interrupt, PEXE shall stop the pseudo-clock by sending a stop message to TAC. If the data collection is run-time and the event was not caused by a breakpoint, the data shall be sent to PMSI. Since PMSI will handle the data and will stop the clock if too much data is sent to it, PEXE shall start the clock and shall then return to IEC. Only IEC can cause an interrupt to PEXE for PMS/DEBUG data collection.

If the event is run-time and is caused by a breakpoint, the data shall be sent to PMSI for processing, but PEXE shall wait to see what the user may now wish to do.

The user may accomplish several types of actions during a breakpoint, which would cause control to return to other places. If the user wishes to continue at this place in PEXE, the clock shall be started; control shall then return to IEC.

If the user had selected to do off-line data collection, the data needed shall be sent through PMSI to buffer files within FCP's memory. After PEXE sends the data to PMSI, PEXE shall start the clock, and shall return control to IEC.

3.2.3.2.4 Error Interrupt

Whenever the hardware or software of VMS finds an error, an interrupt to PEXE will handle the processing of this error. The hardware can find several types of errors: parity error, bus timeout, etc. Software can handle several types: access to non-emulated memory, access to memory not assigned to user, access to module not emulated, etc. Many other types of errors can be controlled for both hardware and software. Each type of error shall be identified by a code. The user shall identify those error types to be active for his run (through RTE). Thus, PEXE shall stop the execution of the total TS when it finds an

active error condition. Then PEXE shall send the error type to RTE, which is executing in the FCP. The user is then told the error code and type of condition by RTE.

3.2.3.2.5 H/W Interval Timer Interrupt

The emulation modules will have one of the 16 Hardware Interval Timers assigned to it. When this module is executed, the real execution time is stored in its Interval Timer. When this Timer decrements to zero, an interrupt is sent to PEXE.

PEXE shall stop the Pseudo-clock while it processes this interrupt. First, PEXE shall determine which Timer caused the interrupt. If Timer 0 caused the interrupt, then all Software Interval Timers are decremented. These exist as a stack of Software Interval Timers which are decremented only when Hardware Timer reaches zero; thus, they are set as multiples of the value stored in Timer 0. There may not be any Software Interval Timers, because these are assigned only when there are not enough Hardware Timers. So, after all Software Interval Timers have been decremented, all are checked to see if any have the value of zero. If so, then the address of the correct handler is found and control shall be transferred there. When control returns from the Handler, that Software Interval Timer is reset and the pseudo-clock shall be restarted.

If Hardware Interval Timer 15 caused the interrupt, then PMS data collection based upon pseudo-time is to be done. Thus, PMS data shall be sent to PMSI and the pseudo-clock shall be restarted.

For all other Hardware Interval Timers, control shall be transferred to the user-assigned Interrupt Handler. Then the pseudo-clock shall be restarted.

In all cases, a default Interrupt Handler shall be part of PEXE. This Handler shall allow the instruction process being

emulated to continue until finished, without further values from the pseudo-clock. Then, when the emulation process is done, control shall return to the step in this phase where the pseudo-clock shall be reset.

Also, the Hardware Timers will be automatically reset. The Software Timers shall be reset by PEEX.

3.2.3.2.6 I/O Request

The hardware emulators for I/O processes can send this type of interrupt when the user wishes to transfer a large block of data from the FCP to some place within the PE for TS usage, or from the TS in the PE to FCP memory.

First, the pseudo-clock shall be stopped. Then the transfer of the data shall be done one word at a time over the MMS. The data can be from memory of the FCP to the TS to emulate input. The data can be from the TS to a memory buffer in the FCP as output. When the total transfer is completed, the pseudo-clock shall be started.

3.2.3.2.7 Scheduler

The Scheduler is a user-selectable routine. The purpose for having the Scheduler is so the user can do I/O by interleaving the accesses with other instruction executions. This interleaving is called cycle stealing. The reading or storing of data shall be one word at a time. When each word process is completed, an interrupt shall be set so that the TS CPU is aware that this process is ready for restart. Meanwhile, control shall be returned to the requesting emulator so that it can continue processing.

3.2.3.2.8 Pseudo-Clock Alignment

A Pseudo-clock Alignment Interrupt is generated whenever the PE's Pseudo-clock is ahead of the Window. The PE

has to remain in an idle state and cannot execute any more macro instructions until the other PE's of the TS can catch up.

If any Bus Messages have been accepted by the BIU of the PE, this Handler shall save the message on a stack (or Buffer). Then the BIU is free to send or receive any other Bus Messages. Also, this Handler shall check to see if the Window is now in alignment with the PE's Pseudo-Time clock. If yes, then control shall be transferred to the IEC so that it can start execution again of a macro instruction. If no, then the Handler shall loop back to again check for Bus Messages.

3.2.3.2.9 Clean Up

When the module of Emulated Instruction Code finds an "End-type" instruction, it sends an interrupt to PEXE.

The pseudo-clock shall be stopped. Whatever is needed by FCP to complete data collection shall be transferred; i.e., tables, ending pseudo-time, etc. Then the "Done" flag shall be sent to the FCP, where final MMS housekeeping is done.

3.2.3.3 Outputs

Because nine phases comprise this package, the outputs were separately identified in paragraphs 3.2.3.2.1 - 3.2.3.2.9.

3.2.4 System Generation Loader or SGL (Module E0)

SGL is illustrated in Figure 3.2.4.

3.2.4.1 Inputs

The user needs to identify whether he wishes to utilize a configured TS which has been saved within the TS library. Else, the user selects one or more already configured Computer Systems and the necessary emulated interconnectors to design a TS. Else, the user selects from the library of emulators (instruction sets and I/O processes) to configure his own TS.

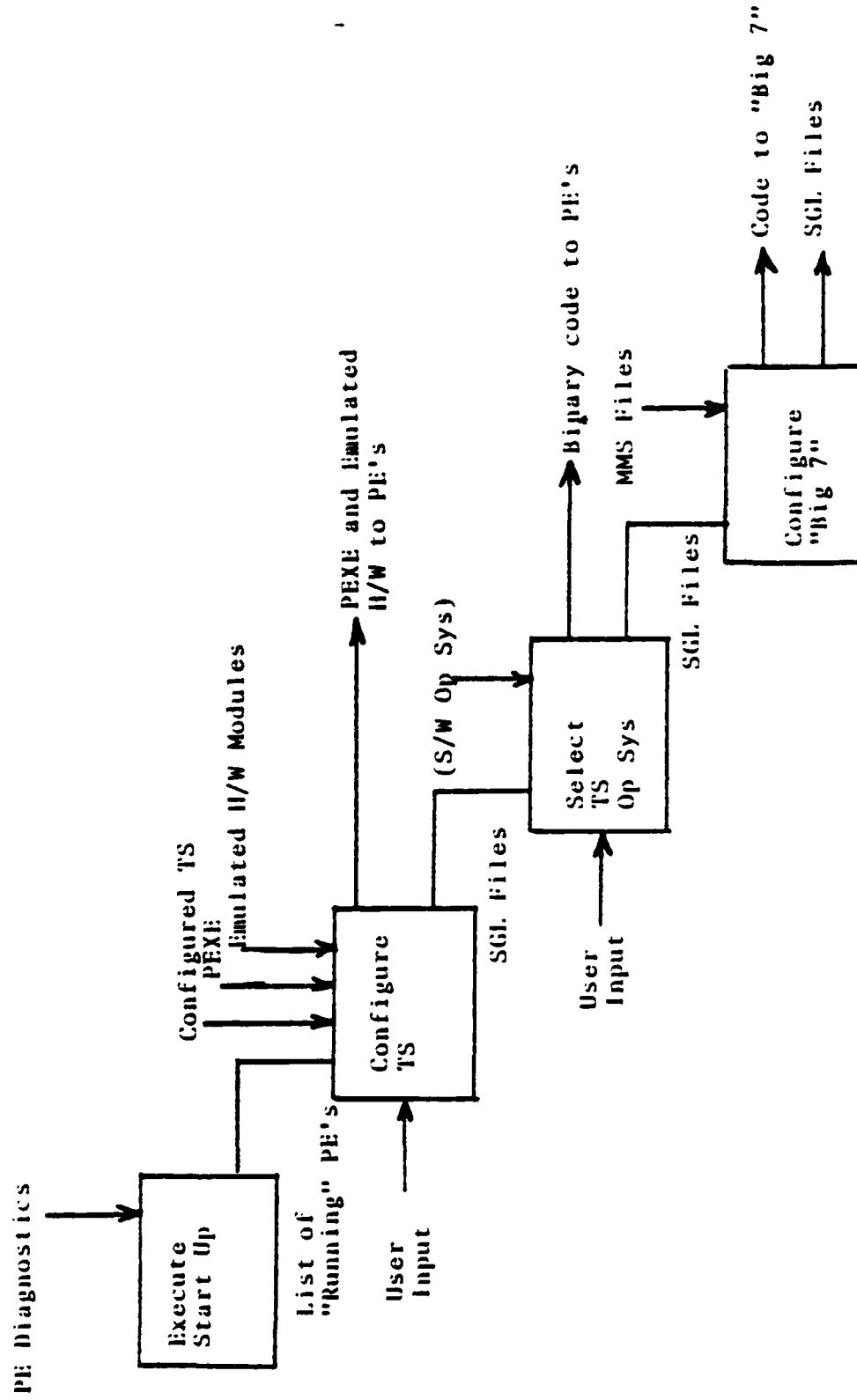


Figure 3.2.4 Top-level Design of SGL

3.2.4.2 Processing

SGL is the first program that the user calls so that he can use MMS. SGL shall use the hardware diagnostic package to verify the total number of PE's (Physical Elements or microprocessors) that are operational. This allows users, who need many PE's for their emulations, to verify that enough PE's are operational.

Then SGL shall allow the user to select:

- A Target System (TS) that has been completely configured, as a network of several computers.
- One or more computer systems that have been completely configured, e.g., an IBM 370/155 and a PDP-11/45.
- Which components he desires from those available in order to configure his own computer system.
- Which interconnects he desires between any emulated computer systems, in order to design his own network system.

A vendor-supplied executive and the emulated modules shall then be downloaded to the PE's by SGL. All linkage shall be accomplished by SGL. When the TS involves a shared resource or needs access to an external device which is emulated on the FCP (Facility Control Processor) computer, SGL shall also do the necessary linkage.

3.2.4.3 Outputs

The output from SGL shall consist of the TS configuration being added to the Library File for future utilization, the MMS hardware being loaded with the emulators of the TS, and a set of SGL Tables to be available for the Run-Time Executive. These SGL Tables shall be mapping tables necessary to link in the TS software and to execute the user programs with, or without, DEBUG/PMS.

3.2.5 Run-Time Executive or RTE (Module F0)

RTE is illustrated in Figure 3.2.5.

3.2.5.1 Inputs

The inputs to RTE are by the user when he

- a. configures the environment, i.e. I/O. The input may be a disk file to provide data, a program to generate data, or a device external to the FCP. The output may be to a disk file or to a device external to the FCP.
- b. selects an operating system and the application programs, which form one absolute module.
- c. selects mode and the necessary parameters to process that mode: straight execution, execution under DEBUG, or execution under PMS.

3.2.5.2 Processing

The Run-Time Executive (RTE) supervises all facets of executing the user's Target System (TS). The Run-Time Executive should be initiated by the user after completion of the System Generation Loader (SGL) phase. The SGL shall initialize and shall ready the Target System for execution. The RTE shall complete the definition of TS I/O; optionally shall configure user-controlled debugging or performance monitoring, and then shall execute the Target System.

The Run-Time Executive software is broken into four phases. The first phase, "Configure Environment" (F1A), shall complete the definition of all Target System I/O by defining buffer memory, shall define the handler software, and shall update tables within the hardware. The "Configure Debug" (F1B) shall allow the user to define debugging options for computers within his Target System. Rather than choosing to DEBUG his Target System, the user may

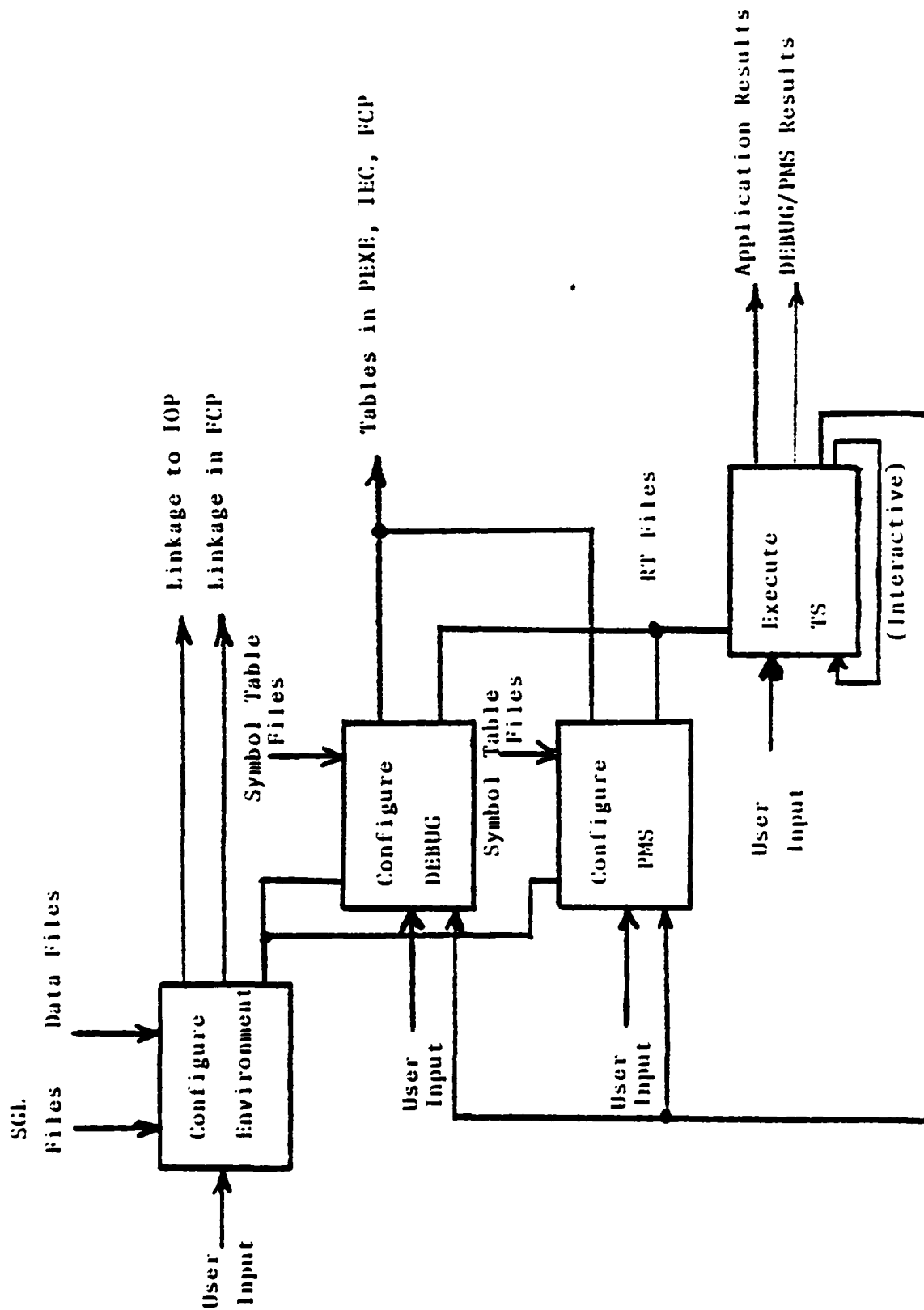


Figure 3.2.5 Top-level Design of RTE

Run-Time
Executive
(RTE)

F Ø

execute under the Performance Monitoring System (PMS). This third phase of RTE, "Configure PMS" (FIC), shall allow the user to select his options for the Performance Monitoring System. The "Execute TS" (FID) shall allow the user to evaluate his emulated system through the interactive use of either DEBUG or PMS.

3.2.5.3 Outputs

The real outputs to the user shall be a combination of:

- a. Applications results, which is the output of the user programs executed on the TS.
- b. DEBUG results, which is the output when the user wishes to verify the instruction-level happenings.
- c. PMS results, which is the output when the user wishes to verify the performance of the TS, i.e. what's happening during emulation.
- d. The DEBUG/PMS can be processed as a combination of run-time and off-line.

3.2.6 Post Mortem Dump or PMD (Module G0)

PMS is illustrated in Figure 3.2.6.

3.2.6.1 Inputs

The first step is to see what the user wishes to do: to terminate or to check contents of some register or memory location, or to do a total dump.

3.2.6.2 Processing

Since the total MMS may be quite large and may thus involve many PE's, the first step shall be a listing of the total system which may be accessed by the user. Thus, the user can access any part of the FCP - as the Program Counter, the registers, memory etc. Any module on the Master Segment, called the

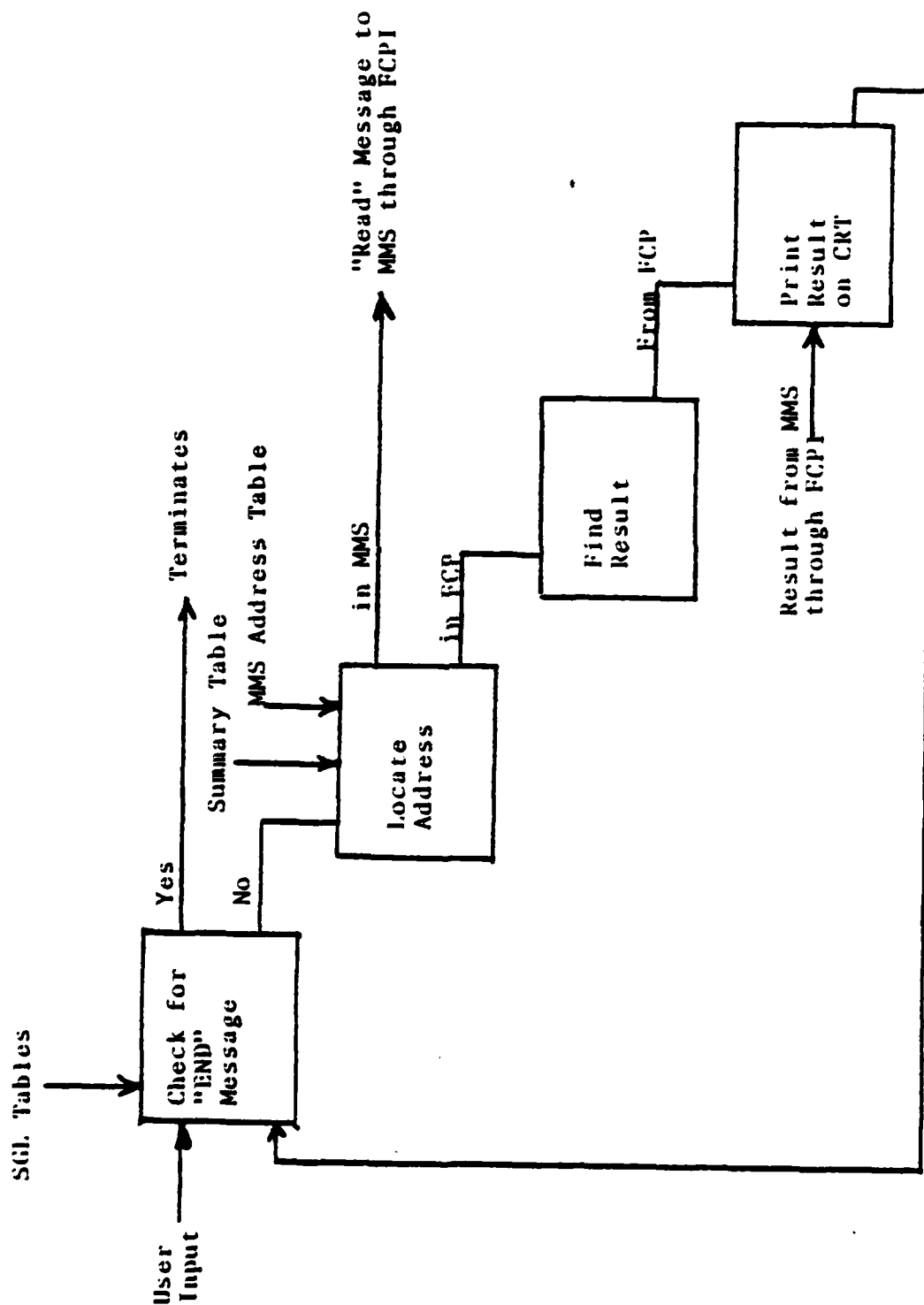


Figure 3.2.6 Top-level Design of PMD

"Big 7", can also be accessed because a table will exist which contains all bus addresses to these modules. The list shall identify all parts of the Computer Systems which are part of the TS, and shall also identify which of the CS parts reside on which PE.

Another option available to the user shall be that the user can dump to disk every register, all memory values, and other addressable locations from all PE's, from the "Big 7", and from the FCP. This shall necessitate much space on the disk. The user could then print out the total dump, as during the night hours, or the user could use the CRT to selectively scan the dump. This allows another user to quickly utilize MMS.

The results from the TS software shall be compared against a macrocode table, and the results shall be printed as the macro instruction along with the numeric value. This numeric value can be octal, decimal, or hexadecimal.

3.2.6.3 Outputs

The output shall be the static values of those areas selected by the user. The output can be dumped to disk, printed on line printer, or displayed on the CRT.

3.2.7 PMS Off-Line Analysis or PMS (Module H0)

PMS is illustrated in Figure 3.2.7

3.2.7.1 Inputs

The user has to select from a menu those routines which he desires to use to select the data, to massage the data, and then to output the results. The library of routines shall include:

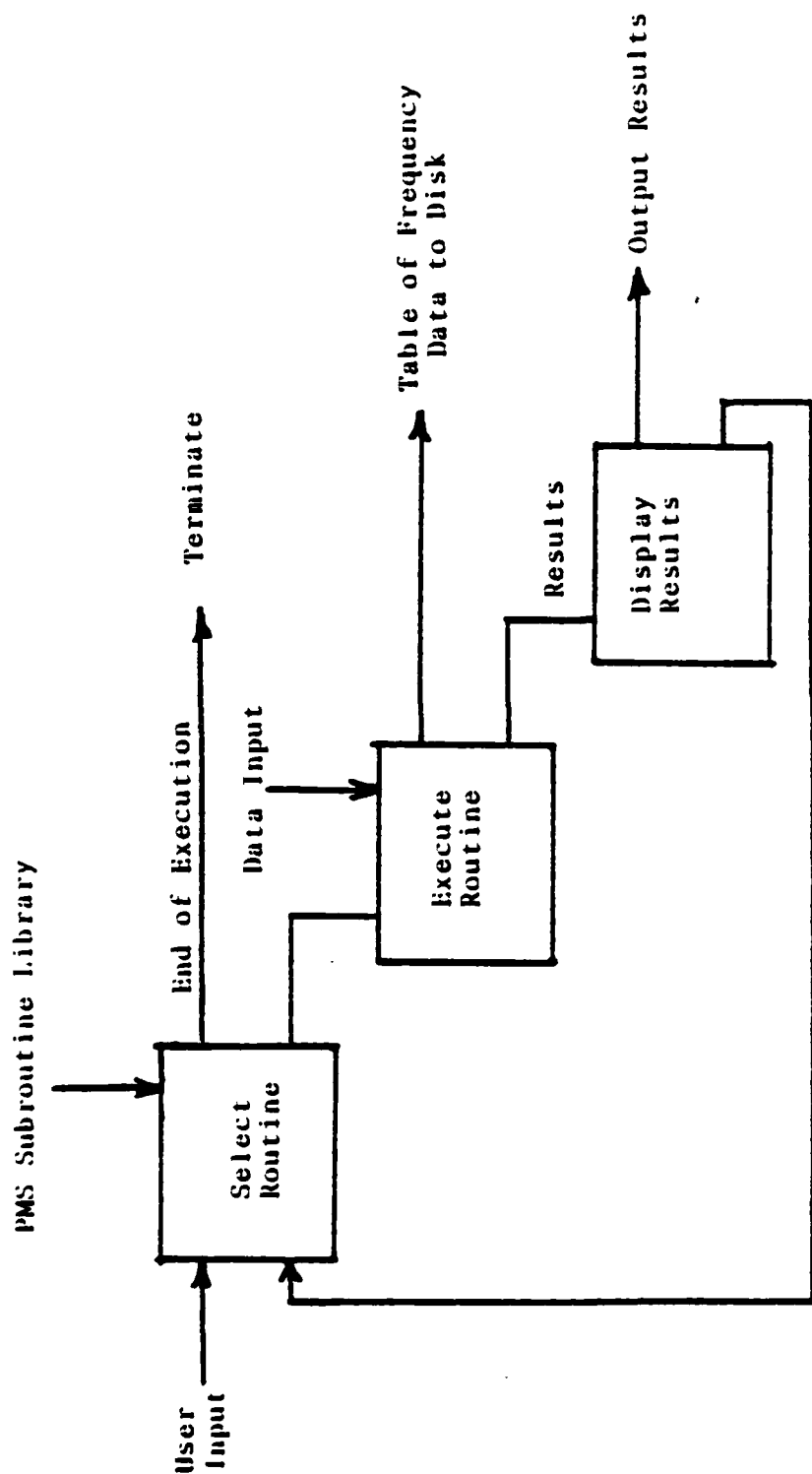


Figure 3.2.7 Top-level Design of PMS

PMS
Off-line
Analysis

HQ

- Display of Data
 - Table printout
 - Scattergram or Scatter Diagram
 - Histogram
 - Table printouts from Statistical Analysis Routines
 - Line fits from fitting data to equations
- Statistical Analysis
 - Frequency Analysis
 - Central Tendency
 - Measure of Variation
- Line Fits
 - Broken Line
 - Straight Line by Selected Points
 - Straight Line by Least Squares
 - Second Degree Parabola by Least Squares
- General Routines
 - Summary of data file, to select entries which match a key
 - Frequency distribution of data file

Once the user has selected the PMS routines to be used, he has to input the name of the data file. He then selects the type of output media, which can be a combination of line printer, CRT or any other device (or disk file).

3.2.7.2 Processing

This package shall be developed as a main program, and as the library of PMS routines selectable from the main program. The user can enhance this library with new routines. Note that this approach of a main routine accessing a library or routines allows the library to be utilized during run-time.

Once the routine is selected, it shall process the data and shall then prepare the output. After processing the results, the user shall have the opportunity to select another process. PMS shall be either interactive, batch mode, or from a command file.

3.2.7.3 Outputs

The output can be either lists, tables, or graphs; the output can be produced on either the line printer or on the CRT.

3.2.8 Target System Emulation or TS (Module IØ)

Figure 3.1d identified the TS which shall be emulated for a test program into MMS. This TS shall consist of the Instruction Emulation Code (IEC) for a PDP-11/45, the emulation of a tape unit, and the emulation of a disk unit. The 11/45 shall be Master of a bus (Bus G), to which to INTEL 8080's are tied as Slaves. The INTEL 8080's shall consist of the IEC instruction set and the emulation of a shared memory. Because of the selected I/O devices, all of the special-purpose PE's on the MMS Master Bus get utilized. Thus, each device indicated needs to be emulated and shall process through the process indicated in Figure 3.2.8.

3.2.8.1 Inputs

The Target System Emulation is a series of inputs by the microcode programmer. These actions shall result in a Symbol Table and microcode object module existing in the host system, which can be linked through SGL with other modules to create a Target System.

This emulation involves eight major steps. These steps are: Identify ID of H /W Emulator, Identify TS User Inputs, Identify Status Words, Identify I/O Buffers, Identify Timing, Create IEC Microcode Module or Create H/W Microcode Module. This shall create the source of the emulation module. The majority of the work required to carry out these functions is done by the micro-

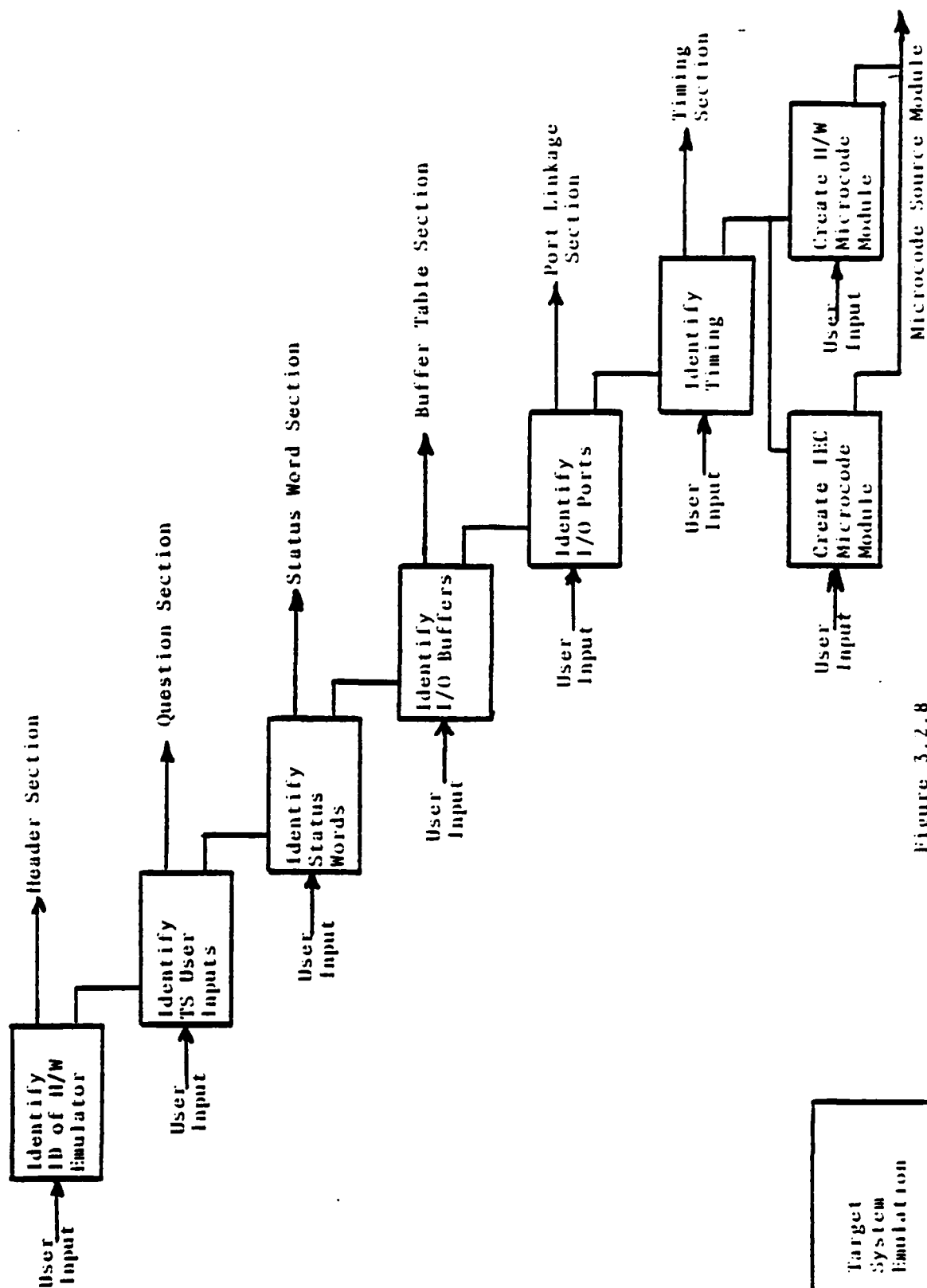


Figure 3.2.8

programmer as input rather than by software.

3.2.8.2 Processing

Once the user has specified the header areas within the first six areas, he then shall input the source (in microcode) which accomplishes the actual emulation. This emulation can be either for an instruction set (Instruction Emulation Code or IEC) or for an I/O process (H/W Microcode Module).

- a. The generation of an IEC microcode module is needed in order to emulate a particular Target System instruction set. This IEC module shall perform many functions. These functions are repeated in sequence: fetch TS instruction from the Load Module, decode instruction to obtain the Opcode, and execute the appropriate microsubroutines. The creator of the IEC shall have to design microcode which shall perform the above operations. This code, in a descriptive format, is appended to the tables that have previously been created.
- b. Hardware microcode modules are used to emulate a particular hardware device. The emulation shall handle actual I/O process as well as the emulation of I/O. The H/W microcode module shall be able to distinguish between the two, to execute the correct microcode, and to store the results in the correct memory or status word address. The microcode shall be written in the same microassembler code as the IEC module. This H/W module is then ready for assembly by the Microassembler.

3.2.8.3 Outputs

The Microassembler shall assemble the entire microcode source module and shall produce a microcode object module and its Symbol Table. This output of the assembler shall be suitable for use by SGL.

3.2.9 Executive for IOP (IOPE)

The IOPE is illustrated in Figure 3.2.9. This executive shall become firmware.

3.2.9.1 Inputs

The inputs to IOPE shall only be from the parts of MMS as a SBS message.

3.2.9.2 Processing

The Executive shall handle the transferring of data for the Target System. During SGL, the user specified what emulators are required, and these are linked to the IOPT, which is an I/O linkage table in IOP. The user has also specified which method is to be used for this data transfer - either DMA or "time slicing". Both methods are part of PEXE.

During RTE, the user specified the "other linkage" to the IOPT. Within FCP, this could be a program, disk file, a mag tape, an external device, etc., to produce input data. Within FCP, the same types could exist to accept data from the TS. Another PE could be also utilized as the "other linkage", and the appropriate addresses would have to be generated and stored within the IOPT.

The Executive, when not busy transferring data, shall scan the IOPT to find the activity with the highest priority. If the activity involves the transfer of a block of data, this amount of data shall be moved. If the data relative to the FCP involves setting of an interrupt, this interrupt shall be set when necessary. When the transfer is completed, a flag shall be set. The entry in the IOPT shall be reset.

The Executive shall wait for a "time out" from the flag in the IOPT for this activity. A signal shall be sent to the PE, and TAC

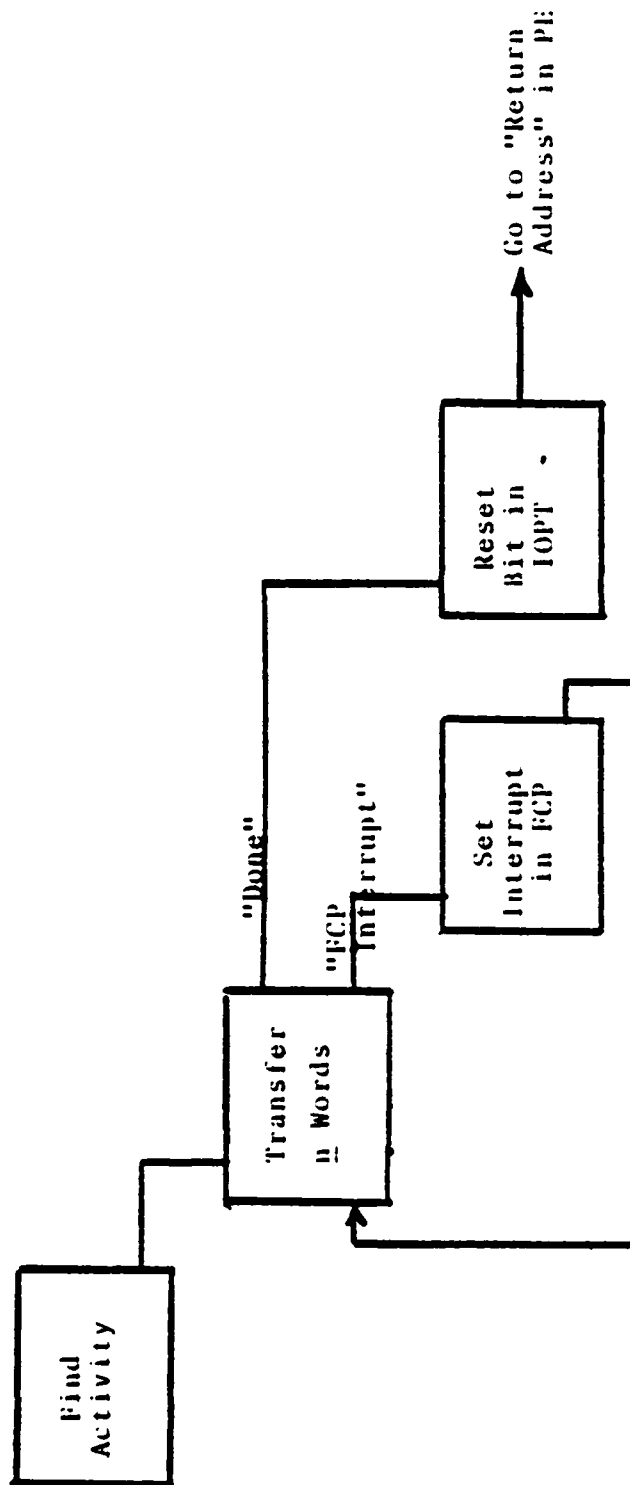


Figure 3.2.9

Executive
for IOP

shall then be started.

3.2.9.3 Outputs

As specified, the output shall consist of data transferred to the destination and an interrupt to be set when indicated to be an interrupt-driven process.

3.2.10 Executive for PMSI (PMSIE)

The PMSIE is illustrated in Figure 3.2.10. This executive shall become firmware.

3.2.10.1 Inputs

The inputs to PMSIE shall only be from the parts of MMS as a SBS message.

3.2.10.2 Processing

The Executive shall first check to see if the event, which caused the PMS/DEBUG data collection process, is a breakpoint. If so, TAC shall be stopped to allow the user to interact with the static condition of the TS. The data shall be sent to a buffer in FCP. An interrupt by PMSI to FCP shall allow the PMS data routines to display the results on the user's terminal. The PMSI Status shall be changed to not busy. Once the user has finished, a continue shall cause PEEX to resume control within the TS.

If the event is not a breakpoint, then the PMSI Executive shall send the data to either one of the run-time/off-line buffers or to both the run-time and off-line buffers. Again, the off-line buffer system shall be a dual-buffer with a utility routine in FCP to handle full buffers. Also, the run-time buffer shall be interrupt-driven; overflow cannot occur here because data is stored over previous data.

Once the data is sent to the FCP, the status of the PMSI Register shall be reset to allow new data to be accepted.

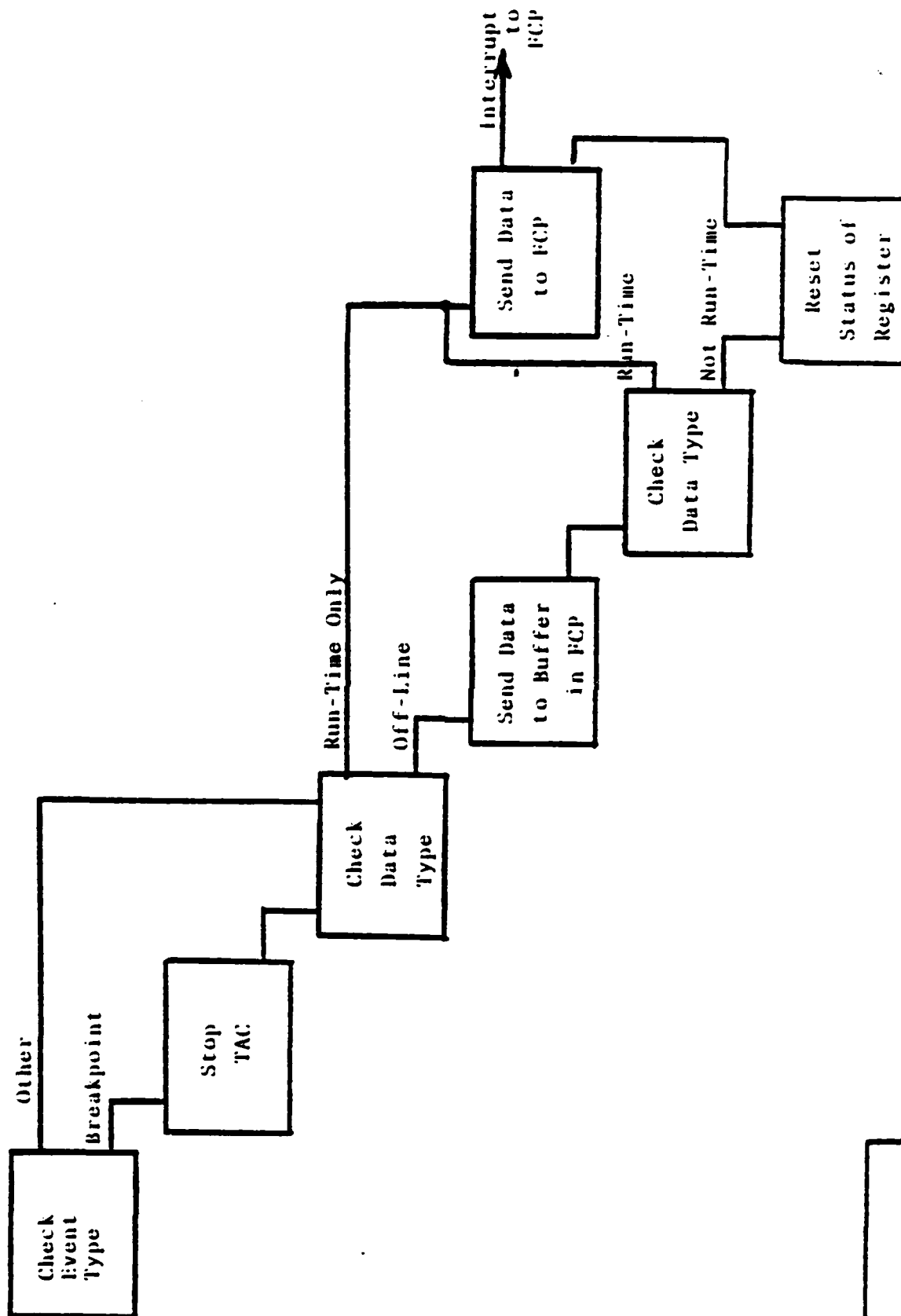


Figure 3.2.10

Executive
for PMSI

3.2.10.3 Outputs

As specified the output shall be DEBUG/PMS data sent by a PE to PMSI, and PMSI sends the data to the correct buffer(s).

3.2.11 Executive for SRC (SRCE)

The SRCE is illustrated in Figure 3.2.11. This executive shall become firmware.

3.2.11.1 Inputs

The inputs to SRCE shall only be from parts of MMS as a SBS message whenever one of the emulated devices wishes to access a shared resource.

3.2.11.2 Processing

The SRC Executive shall resolve arbitration for use of a shared resource. It shall have several ways to obtain the data for the requesting Computer System (CS).

Whenever a CS has requested to use a shared resource (SR), an entry was placed in the Request Table. Therefore, the SRC Executive shall scan this Table for a request of highest priority. If the process for transferring the data is multiplexing with the TS operations, then a data word and the value for arbitration delay shall be sent to the PE. Then the Executive shall wait for the next request from that PE before it sends another data set. Once the PE sends a release message, the Table shall be updated to reflect that this process is finished.

The difference between "burst" mode and "multiplexing" mode is that all data is transferred to the PE's memory, followed by the arbitration time. Again a release message shall be received by SRC.

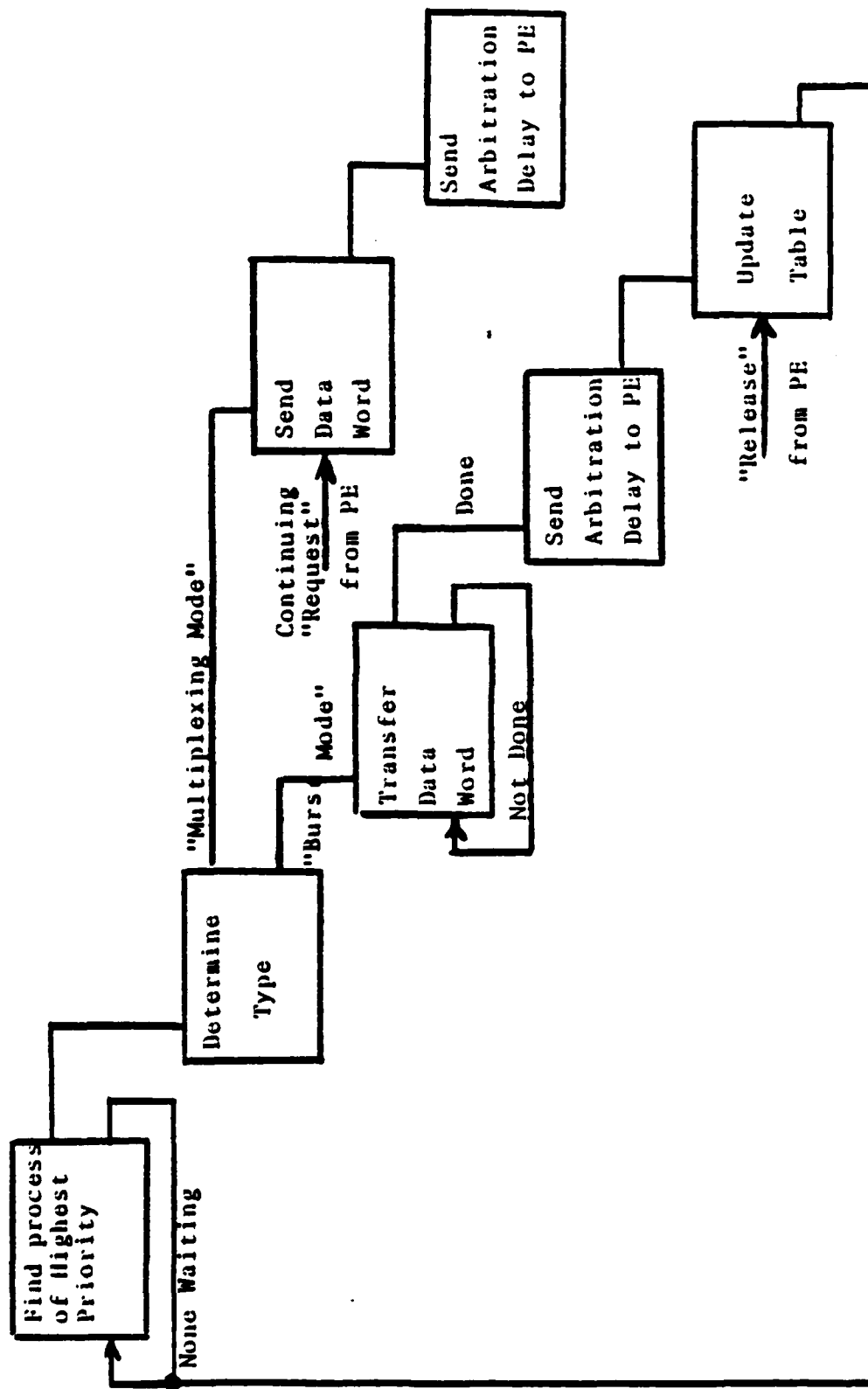


Figure 3.2.11

The SRC Executive shall contain several possible ways to determine the arbitration for the emulated hardware. These methods shall be available to the user as possible schemes to implement.

During "burst" mode, the approach shall be for the hardware to get control and to retain that control until all accesses are satisfied. During emulation, the user shall be able to select zero time, user-specified time, random-time, or the pseudo-time.

3.2.11.3 Outputs

The outputs from SRCE shall be in the form of SBS messages to parts of MMS.

3.3 Adaption

Not applicable.

4.0 QUALITY ASSURANCE PROVISION

4.1 Introduction

In order to test a configuration of special purpose hardware and software as comprises MMS, many tests at various levels are required. The MMS Diagnostic package fulfills the testing at the hardware level. Each major S/W module will need various tests to ascertain correctness of operation. The TS will provide testing at another level - total operation of MMS under control. Thus, some level of testing has already been identified; other tests need to be identified.

4.1.1 Computer Subprogram Testing

All Subprograms shall have testing identified which shall verify that the purpose(s) is (are) being met and that the data in and out of each subprogram has been checked.

4.1.2 Computer Program Testing

All of the identified major modules shall have testing which verifies that all of the specified purposes are being met. Prior to programming, the data into and out of these modules shall have been identified. Tests shall be included as part of the documentation

on each of these modules which identify the correctness of the data in/out. Also, all other testing procedures utilized on the module - level shall be included as part of documentation.

4.1.3 Computer Program Acceptance Testing (CPAT)

The acceptance of a system as large as MMS shall be based partly upon documentation of tests completed. All of these shall be available for CPAT. The exercise of the TS shall be necessary for CPAT. RADC shall also provide test problems to be exercised through MMS. An additional System test shall be the exercising of a TS through ARPANET.

5.0 PREPARATION FOR DELIVERY

Not applicable.

6.0 NOTES

Not applicable.

10.0 APPENDIX I

Not applicable



MISSION of Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C³I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

END

DATE
FILMED

8-80

DTIC